

Accurate Detection.
Real Time Response.
Faster Recovery.

Planned list of Activities and Validation Process for Execution of VAPT

STRYKER, INDIA

Protecting

your business,
your brand, and
everything in between

www.gsecurelabs.com

Planned list of Activities and Validation Process for Execution of VAPT

Warning: THIS DOCUMENT MAY CONTAIN INFORMATION THAT COULD SEVERELY DAMAGE OR IMPACT THE INTEGRITY AND SECURITY OF THE ORGANIZATION IF DISCLOSED PUBLICLY. THIS DOCUMENT SHOULD BE SAFEGUARDED AT ALL TIMES AND MAINTAINED IN A SECURE AREA WHEN NOT IN USE. G' SECURE LABS ASSUMES NO RESPONSIBILITY OR LIABILITY FOR THE SECURITY OF THIS DOCUMENT AFTER DELIVERY TO THE ORGANIZATION NAMED HEREIN. IT IS THE ORGANIZATION'S RESPONSIBILITY TO SAFEGUARD THIS MATERIAL AFTER DELIVERY.

THIS REPORT CONTAINS PROPRIETARY INFORMATION THAT IS NOT TO BE SHARED, COPIED, DISCLOSED OR OTHERWISE DIVULGED WITHOUT THE EXPRESS WRITTEN CONSENT OF G' SECURE LABS OR THEIR DESIGNATED REPRESENTATIVE. USE OF THIS REPORTING FORMAT BY OTHER THAN G' SECURE LABS OR ITS SUBSIDIARIES IS STRICTLY PROHIBITED AND MAY BE PROSECUTED TO THE FULLEST EXTENT OF THE LAW.

Disclaimer: THE RECOMMENDATIONS CONTAINED IN THIS REPORT ARE BASED ON INDUSTRY STANDARD "BEST PRACTICES". BEST PRACTICES ARE, BY NECESSITY, GENERIC IN NATURE AND MAY NOT TAKE INTO ACCOUNT EXACERBATING OR MITIGATING CIRCUMSTANCES. THESE RECOMMENDATIONS, EVEN IF CORRECTLY APPLIED, MAY CAUSE CONFLICTS IN THE OPERATING SYSTEM OR INSTALLED APPLICATIONS. ANY RECOMMENDED CHANGES TO THE OPERATING SYSTEM OR INSTALLED APPLICATION SHOULD FIRST BE EVALUATED IN A NON-PRODUCTION ENVIRONMENT BEFORE BEING DEPLOYED IN THE PRODUCTION ENVIRONMENT.

G' SECURE LABS



Protecting

your business,
your brand, and
everything in between

www.gsecurelabs.com

Recipient:

Name/role	Company
Deepak	Stryker

Document Version:

Name of the Author	Version	Title	Date
Arunesh Mishra	1.0	Planned list of Activities And Validation Process For Execution of VAPT	August 5th, 2022



Table of Contents

A. Summary	6
B. Planned List of Activities and Validation Process.....	6
C. Requirements- Test Cases	63
D. Annexure A.....	65
1. Summary	67
2. Assessment Scope.....	67
3. Assessment Summary	68
3.1 Highlights of Assessment & Testing	68
4. Technical Report of “Smart Medic Tablet Apps and Tablet Device” Project Vulnerability Assessment.....	70
4.1 Smart Medic Tablet App: ICU_V1.0.9-release.apk	71
4.2 Smart Medic Tablet App: Kiosk_V1.0.9-release.apk.....	72
4.3 Smart Medic Tablet App: Management_V1.0.11-release.apk	75
4.4 Smart Medic Tablet App: Settings_V1.0.9-release.apk	76
4.5 Smart Medic Tablet App: SoftwareUpgrade_V1.0.12-release.apk	79
5. Technical Report of “Nurse Station and Smart Medic Azure Application” Project Vulnerability Assessment....	84
4.6 Out-of-band resource load (HTTP)	85
4.7 External service interaction (DNS)	89



4.8 External service interaction (HTTP)	94
4.9 Unencrypted communications.....	100
4.10 Strict transport security not enforced	101
4.11 Email addresses disclosed	104
4.12 Robots.txt file	111
4.13 Open Ports information gathered.....	112
4.14 TLS certificate	115
E. Annexure B.....	117
F. Annexure B.1.....	126
G. Annexure C	145

CONFIDENTIAL



A. Summary

Stryker has assigned the task of carrying out vulnerability assessment and penetration testing of their SmartMedic Platform by G'Secure Labs team. This is planned list of activities and validation process for execution of VAPT task. The version 1.0 result for detailed planned list of activities described about each validation process task which was executed by the Arunesh Mishra and validation process approved by the Harish Shukla and Stryker Team.

The reference of planned list of activities and validation process for execution of VAPT refer the document "Penetration testing (Security testing) Protocol" having document reference number "D001020037 Rev 00".

Only those vulnerabilities which are present during the vulnerability scanning and assessment appear in the report (annexure A), the vulnerabilities which were not found during scan or remediated by development team will not appear in the report. For the reference of the result where the vulnerability was not found, refer the annexure B which contains the list of all the vulnerabilities which were tested by GSL test scanners.

B. Planned List of Activities and Validation Process

Threat Event(s)	Vulnerabilities	Asset	TC No.	Planned list of Activities And Validation Process For Execution of VAPT	Date for Test	Test Executer	Test Result References Topic No.	Result
Deliver undirected malware (CAPEC-185)	Unprotected external USB Port on the tablet/devices.	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	ST No.-01	<ol style="list-style-type: none">1) Create Android malware2) Transfer the malware to tablet/Smart Medic Device3) Malware execution on the device4) Exploit the devices with respect to vulnerability5) Check for the open ports6) Exploit the open ports found while assessment and information gathering.7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1	PASS



Deliver undirected malware (CAPEC-185)	Unprotected external USB Port on the tablet/devices.	Smart medic (Stryker device) System Component	TC No.-02	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1	PASS
Deliver undirected malware (CAPEC-185)	External communications and exposure for communication channels from and to application and devices like tablet and smartmedic device.	Smart medic (Stryker device) System Component	TC No.-03	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Deliver undirected malware (CAPEC-185)	External communications and exposure for communication channels from and	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.-04	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1	PASS



	to application and devices like tablet and smartmedic device.			to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.				
Deliver undirected malware (CAPEC-185)	Legacy system identification if any	Smart medic (Stryker device) System Component	TC No.-05	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Deliver undirected malware (CAPEC-185)	Legacy system identification if any	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.-06	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1	PASS



				7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.				
Deliver undirected malware (CAPEC-185)	Ineffective patch management of firmware, OS and applications throughout the information system plan	Device Maintenance tool (Hardware/Software)	TC No.- 07	NA	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1	PASS
Deliver undirected malware (CAPEC-185)	Ineffective patch management of firmware, OS and applications throughout the information system plan	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 08	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1	PASS
Deliver undirected malware (CAPEC-185)	Ineffective patch management of firmware, OS and applications throughout the information system plan	Smart medic (Stryker device) System Component	TC No.- 09	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				<p>while assessment and information gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>				
Deliver undirected malware (CAPEC-185)	Lack of plan for periodic Software Vulnerability Management	Device Maintenance tool (Hardware/Software)	TC No.- 10	NA	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1	PASS
Deliver undirected malware (CAPEC-185)	Lack of plan for periodic Software Vulnerability Management	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 11	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p> <p>6) Exploit the open ports found while assessment and information gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1	PASS
Deliver undirected malware (CAPEC-185)	Lack of plan for periodic Software Vulnerability Management	Smart medic (Stryker device) System Component	TC No.- 12	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p> <p>6) Exploit the open ports found while assessment and information</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure	PASS



				<p>gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>				
Deliver undirected malware (CAPEC-185)	Unprotected network port(s) on network devices and connection points	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.-13	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p> <p>6) Exploit the open ports found while assessment and information gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1	PASS
Deliver undirected malware (CAPEC-185)	Unprotected network port(s) on network devices and connection points	Smart medic (Stryker device) System Component	TC No.-14	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p> <p>6) Exploit the open ports found while assessment and information gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				8) Exploit the found loopholes while VA scanning using kali tools.				
Deliver undirected malware (CAPEC-185)	Unencrypted data at rest in all possible locations	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 15	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools. 9) Use sniffing tool to sniff the data at motion and MITM	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver undirected malware (CAPEC-185)	Unencrypted data in flight in all flowchannels	Smart medic (Stryker device) System Component	TC No.- 16	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



Deliver undirected malware (CAPEC-185)	Unencrypted data in flight in all flowchannels	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 17	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver undirected malware (CAPEC-185)	Outdated - Software/Hardware	Device Maintenance tool (Hardware/Software)	TC No.- 18	NA	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver undirected malware (CAPEC-185)	Outdated - Software/Hardware	Smart medic (Stryker device) System Component	TC No.- 19	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



Deliver undirected malware (CAPEC-185)	Outdated - Software/Hardware	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 20	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver directed malware (CAPEC-185)	InSecure Configuration for Software/OS on Mobile Devices, Laptops, Workstations, and Servers	Device Maintenance tool (Hardware/Software)	TC No.- 21	NA	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver directed malware (CAPEC-185)	InSecure Configuration for Software/OS on Mobile Devices, Laptops, Workstations, and Servers	Smart medic (Stryker device) System Component	TC No.- 22	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				8) Exploit the found loopholes while VA scanning using kali tools.				
Deliver directed malware (CAPEC-185)	InSecure Configuration for Software/OS on Mobile Devices, Laptops, Workstations, and Servers	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 23	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver directed malware (CAPEC-185)	Unprotected external USB Port on the tablet/devices.	Wireless Network device	TC No.- 24	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Use sniffing tool to sniff the data at motion and MITM 7) Exploit the open ports found while assessment and information gathering. 8) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 9) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



Deliver directed malware (CAPEC-185)	Unprotected external USB Port on the tablet/devices.	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 25	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver directed malware (CAPEC-185)	Unprotected external USB Port on the tablet/devices.	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 26	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Deliver directed malware (CAPEC-185)	External communications and exposure for communication channels from and to application and devices like tablet and smartmedic device.	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 27	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				<p>gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>				
Deliver directed malware (CAPEC-185)	Ineffective patch management of firmware, OS and applications throughout the information system plan	Device Maintenance tool (Hardware/Software)	TC No.- 28	NA	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver directed malware (CAPEC-185)	Ineffective patch management of firmware, OS and applications throughout the information system plan	Smart medic (Stryker device) System Component	TC No.- 29	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p> <p>6) Exploit the open ports found while assessment and information gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Deliver directed malware (CAPEC-185)	Ineffective patch management of firmware, OS and applications throughout the information system plan	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 30	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				<ul style="list-style-type: none"> 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools. 				
Deliver directed malware (CAPEC-185)	Unprotected network port(s) on network devices and connection points	Smart medic (Stryker device) System Component	TC No.- 31	<ul style="list-style-type: none"> 1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Deliver directed malware (CAPEC-185)	Unprotected network port(s) on network devices and connection points	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 32	<ul style="list-style-type: none"> 1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.				
Deliver directed malware (CAPEC-185)	Unprotected network port(s) on network devices and connection points	Wireless Network device	TC No.- 33	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Use sniffing tool to sniff the data at motion and MITM 7) Exploit the open ports found while assessment and information gathering. 8) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 9) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver directed malware (CAPEC-185)	InSecure Configuration for Software/OS on Mobile Devices, Laptops, Workstations, and Servers	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 34	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure A: 5.1, 5.2, 5.3, 5.5, 5.6, 5.7, 5.8, 5.9,	FAIL
Deliver directed malware (CAPEC-185)	InSecure Configuration for Software/OS on Mobile Devices,	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet	TC No.- 35	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



	Laptops, Workstations, and Servers	Application, Network interfaces (Bluetooth, Wifi)		4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.				
Deliver directed malware (CAPEC-185)	Unencrypted data at rest in all possible locations	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 36	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools. 9) Use sniffing tool to sniff the data at motion and MITM	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Deliver directed malware (CAPEC-185)	Unencrypted data at rest in all possible locations	Tablet OS/network details & Tablet Application	TC No.- 37	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				<ul style="list-style-type: none"> 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools. 9) Use sniffing tool to sniff the data at motion and MITM 				
Deliver directed malware (CAPEC-185)	Unencrypted data at rest in all possible locations	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 38	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Use sniffing tool to sniff the data at motion and MITM 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Gaining Access ([S]TRID[E])	Unprotected network port(s) on network devices and connection points	Tablet OS/network details & Tablet Application	TC No.- 39	<ul style="list-style-type: none"> 1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.				
Gaining Access ([S]TRID[E])	Unprotected network port(s) on network devices and connection points	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 40	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Gaining Access ([S]TRID[E])	Unprotected network port(s) on network devices and connection points	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 41	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Gaining Access ([S]TRID[E])	Devices with default passwords needs to be checked for bruteforce attacks	Authentication/Authorisation data	TC No.- 42	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				<ul style="list-style-type: none"> 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 				
Gaining Access ([S]TRID[E])	Devices with default passwords needs to be checked for bruteforce attacks	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 43	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Gaining Access ([S]TRID[E])	Devices with default passwords needs to be checked for bruteforce attacks	Interface/API Communication	TC No.- 44	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



Gaining Access ([S]TRID[E])	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Authentication/Authorisation data	TC No.- 45	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Gaining Access ([S]TRID[E])	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 46	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Gaining Access ([S]TRID[E])	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Smart medic app (Azure Portal Administrator)	TC No.- 47	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.				
Gaining Access ([S]TRID[E])	Checking authentication modes for possible hacks and bypasses	Authentication/Authorisation data	TC No.- 48	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Gaining Access ([S]TRID[E])	Checking authentication modes for possible hacks and bypasses	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 49	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



Gaining Access ([S]TRID[E])	Checking authentication modes for possible hacks and bypasses	Smart medic app (Azure Portal Administrator)	TC No.- 50	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Gaining Access ([S]TRID[E])	Controlled Use of Administrative Privileges over the network	Authentication/Authorisation data	TC No.- 51	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Gaining Access ([S]TRID[E])	Controlled Use of Administrative Privileges over the network	Smart medic app (Azure Portal Administrator)	TC No.- 52	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.				
Gaining Access ([S]TRID[E])	Unprotected external USB Port on the tablet/devices.	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 53	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Maintaining Access (TTP)	Devices with default passwords needs to be checked for bruteforce attacks	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 54	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Maintaining Access (TTP)	Devices with default passwords needs to be	Smart medic app (Azure Portal Administrator)	TC No.- 55	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



	checked for bruteforce attacks			2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.				
Maintaining Access (TTP)	Devices with default passwords needs to be checked for bruteforce attacks	Authentication/Authorisation data	TC No.- 56	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Maintaining Access (TTP)	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 57	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				4) Exploit the open ports using kali tools.				
Maintaining Access (TTP)	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Smart medic app (Azure Portal Administrator)	TC No.- 58	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Maintaining Access (TTP)	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Authentication/Authorisation data	TC No.- 59	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



Maintaining Access (TTP)	Checking authentication modes for possible hacks and bypasses	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 60	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Maintaining Access (TTP)	Checking authentication modes for possible hacks and bypasses	Smart medic app (Azure Portal Administrator)	TC No.- 61	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Maintaining Access (TTP)	Controlled Use of Administrative Privileges over the network	Smart medic app (Azure Portal Administrator)	TC No.- 62	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



Maintaining Access (TTP)	Controlled Use of Administrative Privileges over the network	Authentication/Authorisation data	TC No.- 63	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Clearing Track (TTP)	InSecure Configuration for Software/OS on Mobile Devices, Laptops, Workstations, and Servers	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 64	<ul style="list-style-type: none"> 1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



Clearing Track (TTP)	InSecure Configuration for Software/OS on Mobile Devices, Laptops, Workstations, and Servers	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 65	<p>1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application.</p> <p>2) Check for the open ports using nmap, other kali tools</p> <p>3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.</p> <p>4) Exploit the open ports using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Clearing Track (TTP)	Outdated - Software/Hardware	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 66	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p> <p>6) Exploit the open ports found while assessment and information gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Clearing Track (TTP)	Lack of configuration controls for IT assets in the informaion system plan	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 67	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p> <p>6) Exploit the open ports found while assessment and information</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				<p>gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>				
Clearing Track (TTP)	Lack of configuration controls for IT assets in the information system plan	Device Maintenance tool (Hardware/Software)	TC No.- 68	NA	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Clearing Track (TTP)	Ineffective patch management of firmware, OS and applications throughout the information system plan	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 69	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p> <p>6) Exploit the open ports found while assessment and information gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Clearing Track (TTP)	Ineffective patch management of firmware, OS and applications throughout the information system plan	Device Maintenance tool (Hardware/Software)	TC No.- 70	NA	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



Clearing Track (TTP)	Ineffective patch management of firmware, OS and applications throughout the information system plan	Tablet OS/network details & Tablet Application	TC No.- 71	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Clearing Track (TTP)	The static connection digaram between devices and applications with provision for periodic updaton as per changes	Device Maintainence tool (Hardware/Software)	TC No.- 72	NA	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Clearing Track (TTP)	The static connection digaram between devices and applications with provision for periodic updaton as per changes	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 73	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				8) Exploit the found loopholes while VA scanning using kali tools.				
Elevation of privilege (STRID[E])	Controlled Use of Administrative Privileges over the network	Authentication/Authorisation data	TC No.- 74	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Elevation of privilege (STRID[E])	Controlled Use of Administrative Privileges over the network	Smart medic app (Azure Portal Administrator)	TC No.- 75	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



Denial of service (STRI(D)E)	Unprotected network port(s) on network devices and connection points	Wireless Network device	TC No.- 76	<ul style="list-style-type: none"> 1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Use sniffing tool to sniff the data at motion and MITM 7) Exploit the open ports found while assessment and information gathering. 8) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 9) Exploit the found loopholes while VA scanning using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Denial of service (STRI(D)E)	Unprotected network port(s) on network devices and connection points	Tablet OS/network details & Tablet Application	TC No.- 77	<ul style="list-style-type: none"> 1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



Information disclosure (STR(I)DE)	Unencrypted data at rest in all possible locations	Data at Rest	TC No.- 78	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM 8) Use sniffing tool to sniff the data at motion and MITM 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Information disclosure (STR(I)DE)	Unencrypted data in flight in all flowchannels	Data in Motion	TC No.- 79	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				<p>the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM</p>				
Information disclosure (STR(I)DE)	Weak Encryption Implementaion in data at rest and in motion tactical and design wise	Data at Rest	TC No.- 80	<p>1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Information disclosure (STR(I)DE)	Weak Encryption Implementaion in data at rest and in motion tactical and design wise	Data in Motion	TC No.- 81	<p>1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM				
Information disclosure (STR(I)DE)	Weak Algorithm implementation with respect cipher key size	Data at Rest	TC No.- 82	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



Information disclosure (STR(I)DE)	Weak Algorithm implementation with respect cipher key size	Data in Motion	TC No.- 83	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Information disclosure (STR(I)DE)	InSecure Configuration for Software/OS on Mobile Devices, Laptops, Workstations, and Servers	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 84	<ul style="list-style-type: none"> 1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



Information disclosure (STR(I)DE)	Unencrypted Network segment through the information flow	Data in Motion	TC No.- 85	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Information disclosure (STR(I)DE)	Insecure communications in networks (hospital)	Data in Motion	TC No.- 86	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM				
Data Access (STR[I]DE)	Unprotected network port(s) on network devices and connection points	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 87	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Data Access (STR[I]DE)	Unprotected network port(s) on network devices and connection points	Wireless Network device	TC No.- 88	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Use sniffing tool to sniff the data at motion and MITM 7) Exploit the open ports found while assessment and information gathering. 8) Vulnerability Assessment scanning for the identifying unknown vulnerabilities	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				9) Exploit the found loopholes while VA scanning using kali tools.				
Data Access (STR[I]DE)	Unprotected network port(s) on network devices and connection points	Tablet OS/network details & Tablet Application	TC No.- 89	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Data Access (STR[I]DE)	Devices with default passwords needs to be checked for bruteforce attacks	Data at Rest	TC No.- 90	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				the open port 7) Use sniffing tool to sniff the data at motion and MITM				
Data Access (STR[I]DE)	Devices with default passwords needs to be checked for bruteforce attacks	Authentication/Authorisation data	TC No.- 91	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Data Access (STR[I]DE)	Devices with default passwords needs to be checked for bruteforce attacks	Data in Motion	TC No.- 92	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM				
Data Access (STR[I]DE)	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Data at Rest	TC No.- 93	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Data Access (STR[I]DE)	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Smart medic app (Azure Portal Administrator)	TC No.- 94	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				4) Exploit the open ports using kali tools.				
Data Access (STR[I]DE)	Controlled Use of Administrative Privileges over the network	Smart medic app (Azure Portal Administrator)	TC No.- 95	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Data Access (STR[I]DE)	Unprotected external USB Port on the tablet/devices.	Tablet Resources - web cam, microphone, OTG devices, Removable USB, Tablet Application, Network interfaces (Bluetooth, Wifi)	TC No.- 96	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Open network port exploit (TPP)	Unprotected network port(s) on network devices and connection points	Tablet OS/network details & Tablet Application	TC No.- 97	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools.				
Open network port exploit (TTP)	Unprotected network port(s) on network devices and connection points	Wireless Network device	TC No.- 98	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Use sniffing tool to sniff the data at motion and MITM 7) Exploit the open ports found while assessment and information gathering. 8) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 9) Exploit the found loopholes while VA scanning using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Open network port exploit (TTP)	Unencrypted Network segment through the information flow	Tablet OS/network details & Tablet Application	TC No.- 99	1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



				<p>gathering.</p> <p>7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>8) Exploit the found loopholes while VA scanning using kali tools.</p>				
Open network port exploit (TTP)	Unencrypted Network segment through the information flow	Wireless Network device	TC No.- 100	<p>1) Create Android malware</p> <p>2) Transfer the malware to tablet/Smart Medic Device</p> <p>3) Malware execution on the device</p> <p>4) Exploit the devices with respect to vulnerability</p> <p>5) Check for the open ports</p> <p>6) Use sniffing tool to sniff the data at motion and MITM</p> <p>7) Exploit the open ports found while assessment and information gathering.</p> <p>8) Vulnerability Assessment scanning for the identifying unknown vulnerabilities</p> <p>9) Exploit the found loopholes while VA scanning using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Open network port exploit (TTP)	Controlled Use of Administrative Privileges over the network	Smart medic app (Azure Portal Administrator)	TC No.- 101	<p>1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application.</p> <p>2) Check for the open ports using nmap, other kali tools</p> <p>3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.</p> <p>4) Exploit the open ports using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



Open network port exploit (TTP)	Unencrypted data in flight in all flowchannels	Data in Motion	TC No.- 102	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Open network port exploit (TTP)	Insecure communications in networks (hospital)	Tablet OS/network details & Tablet Application	TC No.- 103	<ul style="list-style-type: none"> 1) Create Android malware 2) Transfer the malware to tablet/Smart Medic Device 3) Malware execution on the device 4) Exploit the devices with respect to vulnerability 5) Check for the open ports 6) Exploit the open ports found while assessment and information gathering. 7) Vulnerability Assessment scanning for the identifying unknown vulnerabilities 8) Exploit the found loopholes while VA scanning using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS



Brute-force Attack (CAPEC-112)	Devices with default passwords needs to be checked for bruteforce attacks	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 104	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	Devices with default passwords needs to be checked for bruteforce attacks	Smart medic app (Azure Portal Administrator)	TC No.- 105	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	Devices with default passwords needs to be checked for bruteforce attacks	Azure Cloud DataBase	TC No.- 106	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port				
Brute-force Attack (CAPEC-112)	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 107	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Smart medic app (Azure Portal Administrator)	TC No.- 108	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Azure Cloud DataBase	TC No.- 109	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port				
Brute-force Attack (CAPEC-112)	Checking authentication modes for possible hacks and bypasses	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 110	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	Checking authentication modes for possible hacks and bypasses	Smart medic app (Azure Portal Administrator)	TC No.- 111	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



Brute-force Attack (CAPEC-112)	Weak Encryption Implementaion in data at rest and in motion tactical and design wise	Data at Rest	TC No.- 112	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	Weak Encryption Implementaion in data at rest and in motion tactical and design wise	Data in Motion	TC No.- 113	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				attempt for the authentication/authorization on the open port 7) Use sniffing tool to sniff the data at motion and MITM				
Social Engineering (TTP)	InSecure Configuration for Software/OS on Mobile Devices, Laptops, Workstations, and Servers	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 114	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B.1, B	PASS
Social Engineering (TTP)	Legacy system identification if any	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 115	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Social Engineering (TTP)	The password complexity or location vulnerability. Like weak passwords and hardcoded passwords.	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 116	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				Burpsuite, kali tools. 4) Exploit the open ports using kali tools.				
Social Engineering (TTP)	Checking authentication modes for possible hacks and bypasses	Interface/API Communication	TC No.- 117	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Social Engineering (TTP)	Checking authentication modes for possible hacks and bypasses	Smart medic app (Stryker Azure Cloud Web Application)	TC No.- 118	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Social Engineering (TTP)	Checking authentication modes for possible hacks and bypasses	Azure Cloud DataBase	TC No.- 119	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port				
Social Engineering (TTP)	Checking authentication modes for possible hacks and bypasses	Smart medic app (Azure Portal Administrator)	TC No.- 120	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Lack of evidence to conclude any malicious attempt/attack (ST[R]IDE)	Insufficient Logging information	Smart medic app (Azure Portal Administrator)	TC No.- 121	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Lack of evidence to conclude any malicious	Insufficient Access permissions for accessing and modifying Log files	Smart medic app (Azure Portal Administrator)	TC No.- 122	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



attempt/attack (ST[R]IDE)				2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.				
Unauthorized Alterations (S[T]RIDE)	InSecure Configuration for Software/OS on Mobile Devices, Laptops, Workstations, and Servers	Azure Cloud DataBase	TC No.- 123	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Unauthorized Alterations (S[T]RIDE)	Insufficient Access permissions for accessing and modifying Log files	Azure Cloud DataBase	TC No.- 124	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				<p>tools.</p> <p>5) Brute Force attempt for the authentication/authorization on the open port</p> <p>6) Exploit related to Brute Force attempt for the authentication/authorization on the open port</p>				
Gaining Access ([S]TRID[E])	Error Info containing sensitive data for Failed Authentication attempts	Smart medic app (Azure Portal Administrator)	TC No.- 125	<p>1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application.</p> <p>2) Check for the open ports using nmap, other kali tools</p> <p>3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.</p> <p>4) Exploit the open ports using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Gaining Access ([S]TRID[E])	Error Info containing sensitive data for Failed Authentication attempts	Azure Cloud DataBase	TC No.- 126	<p>1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application.</p> <p>2) Check for the open ports using nmap, other kali tools</p> <p>3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.</p> <p>4) Exploit the open ports using kali tools.</p> <p>5) Brute Force attempt for the authentication/authorization on the open port</p> <p>6) Exploit related to Brute Force attempt for the</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				authentication/authorization on the open port				
Gaining Access ([S]TRID[E])	Absence of additional security factor along with user identification	Smart medic app (Azure Portal Administrator)	TC No.- 127	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Gaining Access ([S]TRID[E])	Absence of additional security factor along with user identification	Azure Cloud DataBase	TC No.- 128	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Gaining Access ([S]TRID[E])	Having no limit on the login attempts	Smart medic app (Azure Portal Administrator)	TC No.- 129	<ul style="list-style-type: none"> 1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.				
Gaining Access ([S]TRID[E])	Having no limit on the login attempts	Azure Cloud DataBase	TC No.-130	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	Error Info containing sensitive data for Failed Authentication attempts	Smart medic app (Azure Portal Administrator)	TC No.-131	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				4) Exploit the open ports using kali tools.				
Brute-force Attack (CAPEC-112)	Error Info containing sensitive data for Failed Authentication attempts	Azure Cloud DataBase	TC No.- 132	<p>1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application.</p> <p>2) Check for the open ports using nmap, other kali tools</p> <p>3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.</p> <p>4) Exploit the open ports using kali tools.</p> <p>5) Brute Force attempt for the authentication/authorization on the open port</p> <p>6) Exploit related to Brute Force attempt for the authentication/authorization on the open port</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	Absence of additional security factor along with user identification	Smart medic app (Azure Portal Administrator)	TC No.- 133	<p>1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application.</p> <p>2) Check for the open ports using nmap, other kali tools</p> <p>3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools.</p> <p>4) Exploit the open ports using kali tools.</p>	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	Absence of additional security factor along with user identification	Azure Cloud DataBase	TC No.- 134	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port				
Brute-force Attack (CAPEC-112)	Having no limit on the login attempts	Smart medic app (Azure Portal Administrator)	TC No.- 135	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali tools.	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS
Brute-force Attack (CAPEC-112)	Having no limit on the login attempts	Azure Cloud DataBase	TC No.- 136	1) Vulnerability Assessment scanning for the identifying unknown vulnerabilities of web application. 2) Check for the open ports using nmap, other kali tools 3) Exploit the found loopholes while VA Scanning using the Burpsuite, kali tools. 4) Exploit the open ports using kali	July 28th– August 5th, 2022	Arunesh Mishra	Annexure B	PASS



				tools. 5) Brute Force attempt for the authentication/authorization on the open port 6) Exploit related to Brute Force attempt for the authentication/authorization on the open port				
--	--	--	--	---	--	--	--	--

C. Requirements- Test Cases

Test Case No.	Test Case Name	Asset	Test Result References
REQ-STC-001	The Application shall support the use of anti-malware mechanism	Tablet	Annexure C
REQ-STC-002	The Application shall provide secure tunnel Communications channel	Tablet SM Device NSA	Annexure C
REQ-STC-003	Only Stryker made/authenticated devices should be able to communicate with SM device and tablet.	Tablet SM Device	Annexure C
REQ-STC-004	The application shall be allowed to upgrade the tablet application.	Admin	Annexure C
REQ-STC-005	Application shall have the User Management Screen to configure and manage the users as per the roles.	Admin	Annexure C
REQ-STC-006	The Application shall be validated by using invisible captcha during login.	NSA	Annexure C
REQ-STC-007	Invalid login creds, only 3 attempts left.	Admin NSA	Annexure C
REQ-STC-008	System shall store patient id in anonymized fashion.	Admin	Annexure C
REQ-STC-009	Error msg: Something went wrong with API operation try again / contact API admin.	Admin NSA	Annexure C
REQ-STC-010	For sensitive data at rest, identification & proper encryption mechanism needs to be designed & implemented	Tablet Cloud db	Annexure C
REQ-STC-011	Secure sensitive data in the channel flow using strong encryption	Tablet Admin	Annexure C
REQ-STC-012	Weak algorithms such as DES, RC4, etc.. should be avoided and usage of strong algorithms such as AES, RSA, etc.. are recommended	Tablet Cloud db	Annexure C
REQ-STC-013	Typical key lengths are 128 and 256 bits for private keys and 2048 for public keys are recommended.	Tablet Cloud db	Annexure C
REQ-STC-014	If database access using keys/certificates, their generation & storage should be done securely.	Tablet Cloud db	Annexure C



REQ-STC-015	Weight and patient data in the database shall be available upto 6 months.	NSA	Annexure C
REQ-STC-016	User shall find the overall system flow in the User Manual document.	NSA	Annexure C
REQ-STC-017	The application will be used by the Stryker Installation and service teams.	Admin	Annexure C

CONFIDENTIAL



D. Annexure A

Only those vulnerabilities which are present during the vulnerability scanning and assessment appear in the report (annexure A), the vulnerabilities which were not found during scan or remediated by development team will not appear in the report.

Recipient:

Name/role	Company
Deepak	Stryker

Document Version:

Name of the Author	Version	Title	Date
Arunesh Mishra	1.0	Assessment Report	August 5 th , 2022
Sajid Sheikh	1.0	Assessment Report	August 5 th , 2022



Protecting

Table of Contents

1. Summary	67
2. Assessment Scope.....	67
3. Assessment Summary	68
3.1 Highlights of Assessment & Testing	68
4. Technical Report of “Smart Medic Tablet Apps and Tablet Device” Project Vulnerability Assessment.....	70
4.1 Smart Medic Tablet App: ICU_V1.0.9-release.apk	71
4.2 Smart Medic Tablet App: Kiosk_V1.0.9-release.apk.....	72
4.3 Smart Medic Tablet App: Management_V1.0.11-release.apk	75
4.4 Smart Medic Tablet App: Settings_V1.0.9-release.apk	76
4.5 Smart Medic Tablet App: SoftwareUpgrade_V1.0.12-release.apk	79
5. Technical Report of “Nurse Station and Smart Medic Azure Application” Project Vulnerability Assessment....	84
4.6 Out-of-band resource load (HTTP)	85
4.7 External service interaction (DNS)	89
4.8 External service interaction (HTTP)	94
4.9 Unencrypted communications.....	100
4.10 Strict transport security not enforced	101
4.11 Email addresses disclosed	104
4.12 Robots.txt file	111



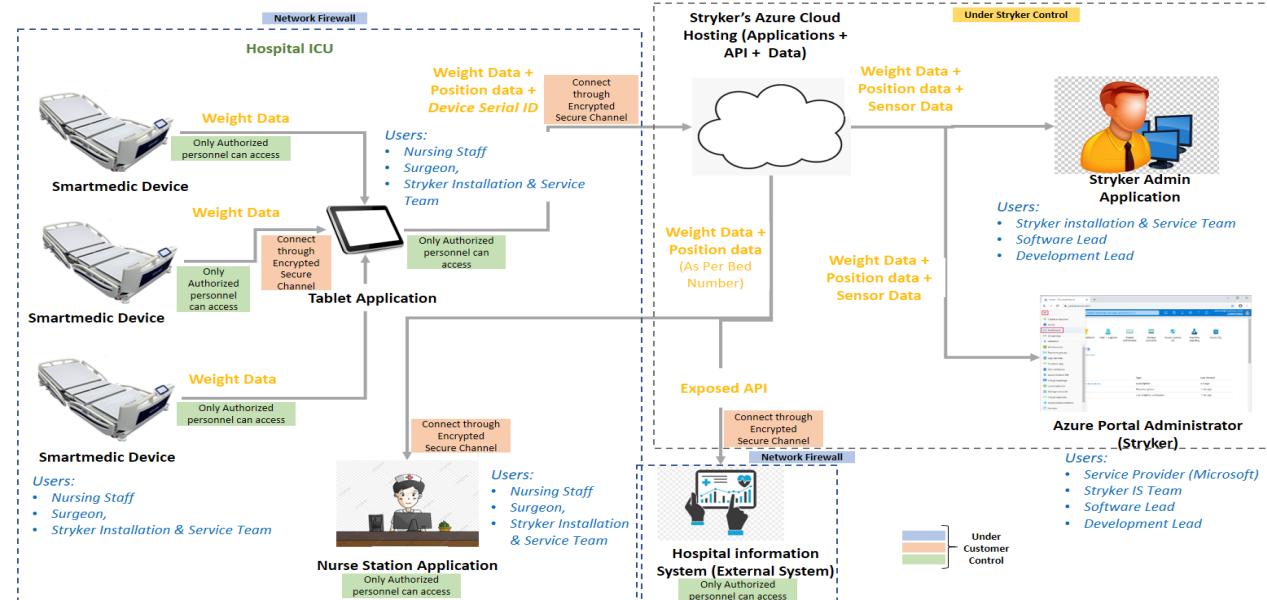
4.13 Open Ports information gathered.....	112
4.14 TLS certificate	115
E.Annexure B.....	117
F.Annexure C.....	145

1. Summary

Stryker has assigned the task of carrying out vulnerability assessment and penetration testing of their Smart Medic Solution by G'Secure Labs team. This task was performed from January 1st, 2022 to August 5th, 2022. The version 1.0 detailed report described about each task and our findings on basis of last build of the solution.

2. Assessment Scope

Vulnerability assessment and penetration testing for Smart Medic Solution was carried for Stryker at following locations – INDIA.



Assets in Scope:

Sr. No.	Smart Medic Components Under Scope
1	Nurse Station Application
2	Smart Medic Azure Applications
3	Smart Medic Tablet Applications
4	Smart Medic Tablet Device
Sr. No.	Urls and Apis Links Related to NSA and Azure Applications
1	https://smartmedic-testing.com
2	https://api.smartmedic-testing.com
3	https://admin.smartmedic-testing.com
Sr. No.	Smart Medic Tablet Applications
1	ICU_V1.0.9-release.apk
2	Kiosk_V1.0.9-release.apk
3	Management_V1.0.11-release.apk
4	Settings_V1.0.9-release.apk
5	SoftwareUpgrade_V1.0.12-release.apk

3. Assessment Summary

3.1 Highlights of Assessment & Testing

G'Secure Labs conducted various scans and tests under authorized white Box test mode on all public & private facing critical assets in the given infrastructure.

The scan report contains the details of the affected applications, systems and assets only and the details of the assets marked as 'safe' has been omitted for the brevity and preciseness of the report. G'Secure Labs used IAST to reduce false positives.

We recommend to address the test reports marked under severity levels 'Critical', 'High' and medium at the earliest and the other severity levels can be taken for discussion and remediation planning. The tests revealed that notable vulnerabilities.

Smart Medic Tablet Device and Apps

- External Storage Accessing

Smart Medic Web Applications

- Strict transport security not enforced
- Email addresses disclosed
- Open Ports Information gathered
- TLS certificate



Accurate **Detection**.
Real Time **Response**.
Faster **Recovery**.



Protecting

your business,
your brand, and
everything in between

www.gsecurelabs.com

Smart Medic Tablet

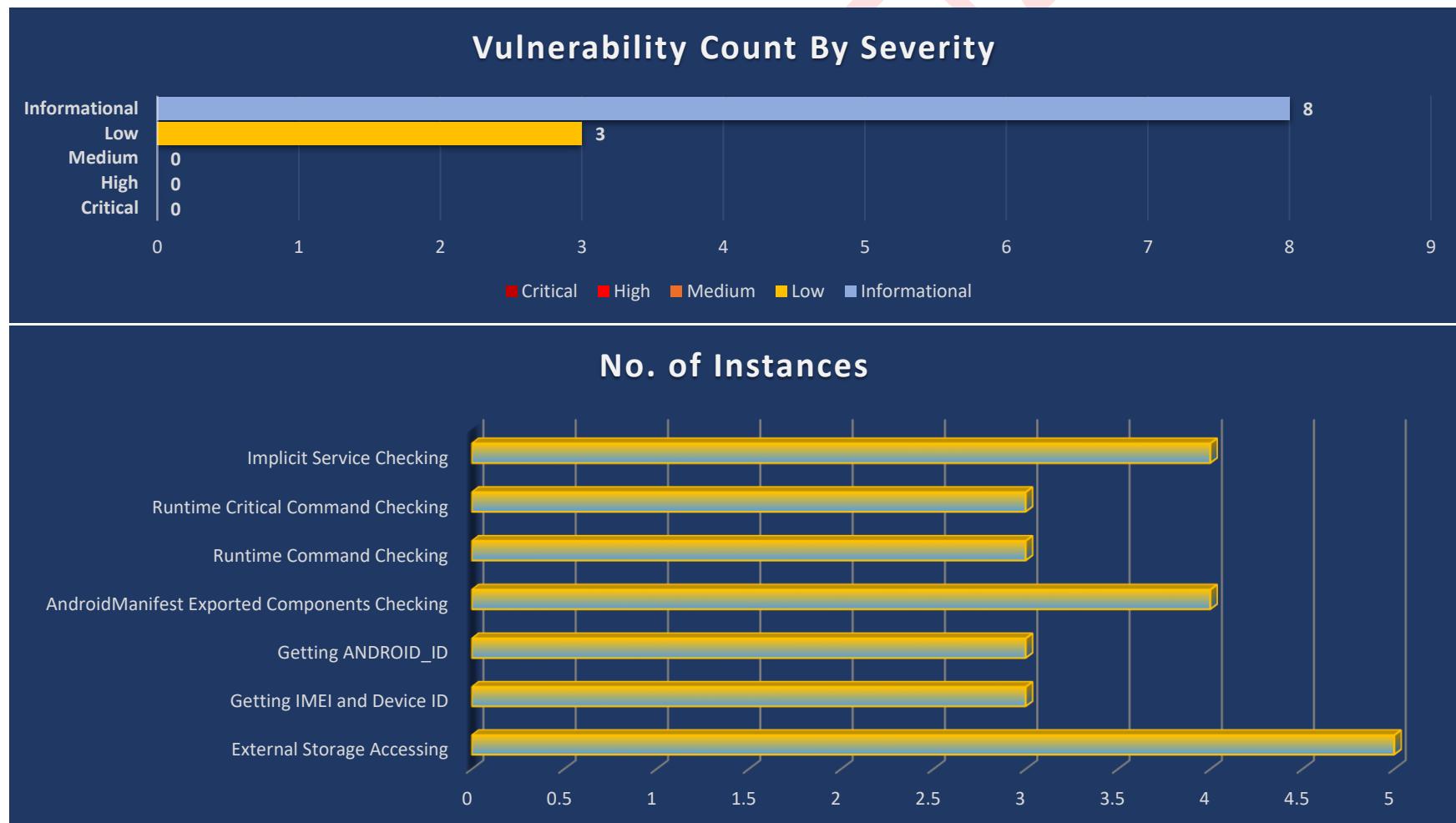
Apps and Tablet Device



4. Technical Report of “Smart Medic Tablet Apps and Tablet Device” Project Vulnerability Assessment

The section uncovers the security checkups, which presents the findings of a security assessment conducted in Smart Medic Tablet Apps (Application Package 1.0.13) and Tablet Device (Testing device no. 1.0.13).

This section reveals the exact areas where your Smart Medic Solution project could be exposed to security threats, areas needed to be improved and it will recommend the solutions to address such risks.



4.1 Smart Medic Tablet App: ICU_V1.0.9-release.apk

Platform: Android

Package Name: com.stryker.icuflow

Package Version Name: 1.0.9

Package Version Code: 9

Min Sdk: 29

Target Sdk: 29

MD5 : 6e8ef0e63d74a3ed1c6afa6a7475655b

SHA1 : 42cef65a8483bfb6b3d6515ab368d9e5e23cfba

SHA256: 613a8f1a2f28ba87a363a7159b0f91f76d688dbff2950e68ccfe23a38f24ca92

SHA512:

8cdc68ba425fe1ea0959cb2b9bd5d6a97ed46dd3ebe55948e0929be5e13b01763503f2ba9663b97b6da0ba9167453d4e63476d78beed717cc6f4d6
b76ec33d28

Analyze Signature:

d4c5105e3d737915d012a82f13ff13836695445bf6b1c24c34b12cfb7ca703429384d4e79129ad54d5eac24ed47818fe821411fa99bf259a7b1613e63
0ace720

Informational Issues:

External Storage Accessing:

External storage access found (Remember DO NOT write important files to external storages):

```
=> Landroidx/core/content/FileProvider;-->parsePathStrategy(Landroid/content/Context;  
    Ljava/lang/String;)Landroidx/core/content/FileProvider$PathStrategy; (0xcc) --->  
    Landroid/os/Environment;-->getExternalStorageDirectory()Ljava/io/File;  
=> Landroidx/core/os/EnvironmentCompat;-->getStorageState(Ljava/io/File;)Ljava/lang/String; (0x34) --->  
    Landroid/os/Environment;-->getExternalStorageDirectory()Ljava/io/File;  
=> Lch/qos/logback/core/android/AndroidContextUtil;-->getExternalStorageDirectoryPath()Ljava/lang/String; (0x0) --->  
    Landroid/os/Environment;-->getExternalStorageDirectory()Ljava/io/File;  
=> Lch/qos/logback/core/android/AndroidContextUtil;-->getMountedExternalStorageDirectoryPath()Ljava/lang/String; (0x2e) --->  
    Landroid/os/Environment;-->getExternalStorageDirectory()Ljava/io/File;  
=> Lcom/stryker/icuflow/utils/Utils$Companion;-->getFilePath()Ljava/lang/String; (0x24) --->  
    Landroid/os/Environment;-->getExternalStorageDirectory()Ljava/io/File;
```

<Sensitive_Information> Getting IMEI and Device ID:

This app has code getting the "device id(IMEI)" but there are problems with this "TelephonyManager.getDeviceId()" approach.



1. Non-phones: Wifi-only devices or music players that don't have telephony hardware just don't have this kind of unique identifier.
2. Persistence: On devices which do have this, it persists across device data wipes and factory resets. It's not clear at all if, in this situation, your app should regard this as the same device.
3. Privilege: It requires READ_PHONE_STATE permission, which is irritating if you don't otherwise use or need telephony.
4. Bugs: We have seen a few instances of production phones for which the implementation is buggy and returns garbage, for example zeros or asterisks.

If you want to get an unique id for the device, we suggest you use "Installation" framework in the following article.

Please check the reference: <http://android-developers.blogspot.tw/2011/03/identifying-app-installations.html>

```
=> Landroidx/core/telephony/TelephonyManagerCompat;=>getImei(Landroid/telephony/TelephonyManager;)Ljava/lang/String;  
(0xb0)  
---> Landroid/telephony/TelephonyManager;=>getDeviceId()Ljava/lang/String;
```

<Sensitive_Information> Getting ANDROID_ID:

This app has code getting the 64-bit number "Settings.Secure.ANDROID_ID".

ANDROID_ID seems a good choice for a unique device identifier. There are downsides: First, it is not 100% reliable on releases of Android prior to 2.2 (Froyo).

Also, there has been at least one widely-observed bug in a popular handset from a major manufacturer, where every instance has the same ANDROID_ID.

If you want to get an unique id for the device, we suggest you use "Installation" framework in the following article.

Please check the reference: <http://android-developers.blogspot.tw/2011/03/identifying-app-installations.html>

```
=> Lcom/stryker/icuflow/utils/AppUtils$Companion;=>getProvisionedId(Landroid/content/Context;)Ljava/lang/String; (0x16) --->  
Landroid/provider/Settings$Secure;=>getString(Landroid/content/ContentResolver; Ljava/lang/String;)Ljava/lang/String;
```

4.2 Smart Medic Tablet App: Kiosk_V1.0.9-release.apk

Platform: Android

Package Name: com.stryker.kioskmode

Package Version Name: 1.0.9

Package Version Code: 9

Min Sdk: 29

Target Sdk: 29

MD5 : 39253543da610a59648f530a79bd7072

SHA1 : 72e9e7918b4cc0550208a3bd060713b16e561f71

SHA256: c7444b5fcfd698547b954ef6bdf79adb63b324395e863b6d94bb72903522dcd



SHA512:

6ff93c5809a28eaefa13b7711abec0e23c03e3635b08b5cbc64b43ca29116ed1b86327852ff0c6db1c89d93f4cbda480dde736691651bf2765d92fe75
5011926

Analyze Signature:

22c0716850d758501151d10fc77097d1e316f392e5d9d2fafb16acac817a3b1f639cb032f71973eb7d88873b4d8ea1e5e05d71891af394af12502800a
072abc3

Low Issues:

<Command> Runtime Command Checking:

This app is using critical function 'Runtime.getRuntime().exec("...")'.

Please confirm these following code sections are not harmful:

```
=>Lcom/stryker/kioskmode/utils/ExtensionsKt;->performSuCommand(Ljava/lang/String;)V (0x34) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=> Lcom/stryker/kioskmode/ui/eula/EulaActivity;->disableGoogleSearch()V (0xc) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=> Lcom/stryker/kioskmode/ui/eula/EulaActivity;->resetSystemConnectionSetting()V (0x10) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=> Lcom/stryker/kioskmode/ui/splash/SplashActivity;->grantPermission()V (0x14) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
```

<Command> Runtime Critical Command Checking:

Requesting for "root" permission code sections 'Runtime.getRuntime().exec("su")' found (Critical but maybe false positive):

```
=>Lcom/stryker/kioskmode/utils/ExtensionsKt;->performSuCommand(Ljava/lang/String;)V (0x34) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=> Lcom/stryker/kioskmode/ui/eula/EulaActivity;->resetSystemConnectionSetting()V (0x10) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=> Lcom/stryker/kioskmode/ui/splash/SplashActivity;->grantPermission()V (0x14) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
```

<Implicit_Intent> Implicit Service Checking:

To ensure your app is secure, always use an explicit intent when starting a Service and DO NOT declare intent filters for your services. Using an implicit intent to start a service is a security hazard because you cannot be certain what service will respond to the intent, and the user cannot see which service starts.

Reference: <http://developer.android.com/guide/components/intents-filters.html#Types>

```
=> com.stryker.kioskmode.service.KioskService
```



Informational Issues:

External Storage Accessing:

External storage access found (Remember DO NOT write important files to external storages):

```
=> Landroidx/core/content/FileProvider;->parsePathStrategy(Landroid/content/Context;
    Ljava/lang/String;)Landroidx/core/content/FileProvider$PathStrategy; (0xcc) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Landroidx/core/os/EnvironmentCompat;->getStorageState(Ljava/io/File;)Ljava/lang/String; (0x34) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lch/qos/logback/core/android/AndroidContextUtil;->getExternalStorageDirectoryPath()Ljava/lang/String; (0x0) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lch/qos/logback/core/android/AndroidContextUtil;->getMountedExternalStorageDirectoryPath()Ljava/lang/String; (0x2e) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lcom/stryker/kioskmode/utils/ExtensionsKt;->readAppListFileOnInternalStorage()Ljava/lang/String; (0x4) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lcom/stryker/kioskmode/utils/ExtensionsKt;->writeAppListFileOnInternalStorage(Ljava/lang/String;)V (0x4) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=> Lcom/stryker/kioskmode/utils/Utils$Companion;->getFilePath()Ljava/lang/String; (0x24) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
```

AndroidManifest Exported Components Checking:

Found "exported" components(except for Launcher) for receiving outside applications' actions (AndroidManifest.xml).

These components can be initialized by other apps. You should add or modify the attribute to [exported="false"] if you don't want to.

You can also protect it with a customized permission with "signature" or higher protectionLevel and specify in "android:permission" attribute.

```
service => com.stryker.kioskmode.service.KioskService
```

<Sensitive_Information> Getting ANDROID_ID:

This app has code getting the 64-bit number "Settings.Secure.ANDROID_ID".

ANDROID_ID seems a good choice for a unique device identifier. There are downsides: First, it is not 100% reliable on releases of Android prior to 2.2 (Froyo).

Also, there has been at least one widely-observed bug in a popular handset from a major manufacturer, where every instance has the same ANDROID_ID.

If you want to get an unique id for the device, we suggest you use "Installation" framework in the following article.

Please check the reference: <http://android-developers.blogspot.tw/2011/03/identifying-app-installations.html>

```
=>Lcom/google/firebase/crashlytics/internal/common/CommonUtils;->isEmulator(Landroid/content/Context;)Z (0xc) --->
```



Landroid/provider/Settings\$Secure; ->getString(Landroid/content/ContentResolver; Ljava/lang/String;)Ljava/lang/String;

4.3 Smart Medic Tablet App: Management_V1.0.11-release.apk

Platform: Android

Package Name: com.stryker.management

Package Version Name: 1.0.11

Package Version Code: 11

Min Sdk: 29

Target Sdk: 29

MD5 : 4cf88538e291c4211b3069ca3d661700

SHA1 : 41d7a8ea5169ae5662ef85a531effb6cb3863241

SHA256: 9ccff517bdab6302fd9b78c3772f94667eaf3d456c9701bb89730cec519d7ae9

SHA512:

701647a29f2a3d56271aab7a16827d89fc5a0eef99b78c341818794368072b819201816f86be61df5f6cca8c00c8947d99a1450adb4b5b76d8daa680
8771f846

Analyze Signature:

28a658c16cc179f3d058308a57c0af4df6284bc8581b1c7acc604ef0fef5a1cc0a98fed9c578f6cefc380cee08dcff253dd2c07eae5cbcd5a324cd3b444f
66fc

Low Issues:

<Implicit_Intent> Implicit Service Checking:

To ensure your app is secure, always use an explicit intent when starting a Service and DO NOT declare intent filters for your services. Using an implicit intent to start a service is a security hazard because you cannot be certain what service will respond to the intent, and the user cannot see which service starts.

Reference: <http://developer.android.com/guide/components/intents-filters.html#Types>

```
=> com.stryker.management.Services.MyBluetoothService
=> com.stryker.management.Services.ServiceForInternetCheck
=> com.stryker.management.Services.ServiceForInternetCheck
=> com.stryker.management.Services.ManagementService
```

Informational Issues:

External Storage Accessing: External storage access found (Remember DO NOT write important files to external storages):



```
=> Landroidx/core/content/FileProvider;->parsePathStrategy(Landroid/content/Context;
    Ljava/lang/String;)Landroidx/core/content/FileProvider$PathStrategy; (0xcc) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Landroidx/core/os/EnvironmentCompat;->getStorageState(Ljava/io/File;)Ljava/lang/String; (0x34) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lch/qos/logback/core/android/AndroidContextUtil;->getExternalStorageDirectoryPath()Ljava/lang/String; (0x0) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lch/qos/logback/core/android/AndroidContextUtil;->getMountedExternalStorageDirectoryPath()Ljava/lang/String; (0x2e) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
```

AndroidManifest Exported Components Checking:

Found "exported" components(except for Launcher) for receiving outside applications' actions (AndroidManifest.xml).

These components can be initialized by other apps. You should add or modify the attribute to [exported="false"] if you don't want to.

You can also protect it with a customized permission with "signature" or higher protection Level and specify in "android:permission" attribute.

```
service => com.stryker.management.Services.MyBluetoothService
service => com.stryker.management.Services.ServiceForInternetCheck
service => com.stryker.management.Services.ManagementService
```

<Sensitive_Information> Getting IMEI and Device ID:

This app has code getting the "device id(IMEI)" but there are problems with this "TelephonyManager.getDeviceId()" approach.

1.Non-phones: Wifi-only devices or music players that don't have telephony hardware just don't have this kind of unique identifier.

2.Persistence: On devices which do have this, it persists across device data wipes and factory resets. It's not clear at all if, in this situation, your app should regard this as the same device.

3.Privilege:It requires READ_PHONE_STATE permission, which is irritating if you don't otherwise use or need telephony.

4.Bugs: We have seen a few instances of production phones for which the implementation is buggy and returns garbage, for example zeros or asterisks.

If you want to get an unique id for the device, we suggest you use "Installation" framework in the following article.

Please check the reference: <http://android-developers.blogspot.tw/2011/03/identifying-app-installations.html>

```
=>Landroidx/core/telephony/TelephonyManagerCompat;->getImei(Landroid/telephony/TelephonyManager;)Ljava/lang/String;
    (0xb0)
---> Landroid/telephony/TelephonyManager;->getDeviceId()Ljava/lang/String;
```

4.4 Smart Medic Tablet App: Settings_V1.0.9-release.apk

Platform: Android

Package Name: com.stryker.settings



Package Version Name: 1.0.9

Package Version Code: 9

Min Sdk: 29

Target Sdk: 29

MD5 : e20051c1f5e179018100ae938304049e

SHA1 : c4b20ad36559431ae46e93798245d2f6dbdf88d2

SHA256: 865ee6886ca60b421d22409fe4cd943de4fec96d0ade469ffc196c9ce0f5a3fb

SHA512:

dccae5321dd721a70346e96d94f0fc93c7e02bc77a6e67e79a1ba1e48edba80a49c0856b97509013f963960ebc5757e7c11d4d862f19eaf64b76c0cb9551d640

Analyze Signature:

cc26ff36d3aa05ccb856f7d431519c766022b915a75a5237cca93d391a2afee6a30ff2d259629affb56b0c9d2c641768447bac5b734c45449afec9117736ba7

Low Issues:

<Command> Runtime Command Checking:

This app is using critical function 'Runtime.getRuntime().exec("...")'.

Please confirm these following code sections are not harmful:

```
=>Lcom/stryker/settings/utils/Utils$Companion;->performSuCommand(Ljava/lang/String;)V (0x36) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=> Lcom/stryker/settings/utils/WifiSettings;->getTetheringConfig()Lkotlin/Pair; (0x10) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=>Lcom/stryker/settings/utils/WifiSettings;->getWifiConfig(Landroid/content/Context;)Lkotlin/Pair; (0x24) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=>Lcom/stryker/settings/utils/WifiSettings;->getWifiConfig(Landroid/content/Context;)Lkotlin/Pair; (0x40) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=>Lcom/stryker/settings/ui/diagnostics/DiagnosticsViewModel$logCatOutput$1;->invokeSuspend(Ljava/lang/Object;)Ljava/lang/Object; (0xb0) ---> Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=>Lcom/stryker/settings/ui/diagnostics/DiagnosticsViewModel$logCatOutput$1;->invokeSuspend(Ljava/lang/Object;)Ljava/lang/Object; (0xde) ---> Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=>Lcom/stryker/settings/ui/connections/ConnectionsActivity;->checkOverlayPermission()V (0x1e) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
```

<Command> Runtime Critical Command Checking:

Requesting for "root" permission code sections 'Runtime.getRuntime().exec("su")' found (Critical but maybe false positive):



```
=>Lcom/stryker/settings/utils/Utils$Companion;->performSuCommand(Ljava/lang/String;)V (0x36) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=>Lcom/stryker/settings/utils/WifiSettings;->getWifiConfig(Landroid/content/Context;)Lkotlin/Pair; (0x24) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
```

<Implicit_Intent> Implicit Service Checking:

To ensure your app is secure, always use an explicit intent when starting a Service and DO NOT declare intent filters for your services. Using an implicit intent to start a service is a security hazard because you cannot be certain what service will respond to the intent, and the user cannot see which service starts.

Reference: <http://developer.android.com/guide/components/intents-filters.html#Types>

```
=> com.stryker.settings.services.SettingsService
```

Informational Issues:

External Storage Accessing:

External storage access found (Remember DO NOT write important files to external storages):

```
=> Landroidx/core/content/FileProvider;->parsePathStrategy(Landroid/content/Context;
    Ljava/lang/String;)Landroidx/core/content/FileProvider$PathStrategy; (0xcc) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Landroidx/core/os/EnvironmentCompat;->getStorageState(Ljava/io/File;)Ljava/lang/String; (0x34) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lch/qos/logback/core/android/AndroidContextUtil;->getExternalStorageDirectoryPath()Ljava/lang/String; (0x0) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lch/qos/logback/core/android/AndroidContextUtil;->getMountedExternalStorageDirectoryPath()Ljava/lang/String; (0x2e) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=> Lcom/stryker/settings/utils/Utils$Companion;->getFilePath()Ljava/lang/String; (0x24) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
```

AndroidManifest Exported Components Checking:

Found "exported" components(except for Launcher) for receiving outside applications' actions (AndroidManifest.xml).

These components can be initialized by other apps. You should add or modify the attribute to [exported="false"] if you don't want to.

You can also protect it with a customized permission with "signature" or higher protectionLevel and specify in "android:permission" attribute.

```
service => com.stryker.settings.services.SettingsService
```



<Sensitive_Information> Getting ANDROID_ID:

This app has code getting the 64-bit number "Settings.Secure.ANDROID_ID".

ANDROID_ID seems a good choice for a unique device identifier. There are downsides: First, it is not 100% reliable on releases of Android prior to 2.2 (Froyo).

Also, there has been at least one widely-observed bug in a popular handset from a major manufacturer, where every instance has the same ANDROID_ID.

If you want to get an unique id for the device, we suggest you use "Installation" framework in the following article.

Please check the reference: <http://android-developers.blogspot.tw/2011/03/identifying-app-installations.html>

```
=>Lcom/google/firebase/crashlytics/internal/common/CommonUtils;=>isEmulator(Landroid/content/Context;)Z (0xc) --->
    Landroid/provider/Settings$Secure;=>getString(Landroid/content/ContentResolver; Ljava/lang/String;)Ljava/lang/String;

=>Lcom/stryker/settings/utils/TabletVersionUtils$Companion;=>getProvisionedId(Landroid/content/Context;)Ljava/lang/String;
    (0x16) ---> Landroid/provider/Settings$Secure;=>getString(Landroid/content/ContentResolver;
    Ljava/lang/String;)Ljava/lang/String;
```

4.5 Smart Medic Tablet App: SoftwareUpgrade_V1.0.12-release.apk

Platform: Android

Package Name: com.stryker.softwareupgrade

Package Version Name: 1.0.12

Package Version Code: 12

Min Sdk: 29

Target Sdk: 29

MD5 : c438308f8469ccd12a63b1bca43b23ec

SHA1 : 058d046c0bbe60b623dd0444d8fb349eca94a96b

SHA256: 58713b8091e501d49669d7e63c450edb5bc5178923921c4352d530eeae611c64

SHA512:

ec2a87503fc3972f51463edc89bd881374e4d5d7c99eb6103e1d31ab28b52b20ab79f0999e5212ba05e30f30d92b383a5791029bd36b320b8e1e55d
8bf45f311

Analyze Signature:

66b1be1c4f26088224ee7ee62d72fa1e84b3efa07e924cf55edc39bd3ab0b055e65c532609386b91e2a0cca719b24d211aefe195972b0004f4ecd9f7d
41cd317

Low Issues:



<Command> Runtime Command Checking:

This app is using critical function 'Runtime.getRuntime().exec("...")'.

Please confirm these following code sections are not harmful:

```
=>Lcom/stryker/softwareupgrade/services/CheckIsAllServiceRunning;->performAdbCommand(Ljava/lang/String;)V (0x14) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=>Lcom/stryker/softwareupgrade/ui/SoftwareUpgradeActivity;->androidSilentInstallApk(Ljava/lang/String; Ljava/lang/String;)V
    (0x74) ---> Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
```

<Command> Runtime Critical Command Checking:

Requesting for "root" permission code sections 'Runtime.getRuntime().exec("su")' found (Critical but maybe false positive):

```
=>Lcom/stryker/softwareupgrade/services/CheckIsAllServiceRunning;->performAdbCommand(Ljava/lang/String;)V (0x14) --->
    Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
=>Lcom/stryker/softwareupgrade/ui/SoftwareUpgradeActivity;->androidSilentInstallApk(Ljava/lang/String; Ljava/lang/String;)V
    (0x74) ---> Ljava/lang/Runtime;->exec(Ljava/lang/String;)Ljava/lang/Process;
```

<Implicit_Intent> Implicit Service Checking:

To ensure your app is secure, always use an explicit intent when starting a Service and DO NOT declare intent filters for your services. Using an implicit intent to start a service is a security hazard because you cannot be certain what service will respond to the intent, and the user cannot see which service starts.

Reference: <http://developer.android.com/guide/components/intents-filters.html#Types>

```
=> com.stryker.softwareupgrade.services.SoftwareUpgradeService
```

Informational Issues:

External Storage Accessing:

External storage access found (Remember DO NOT write important files to external storages):

```
=> Landroidx/core/content/FileProvider;->parsePathStrategy(Landroid/content/Context;
    Ljava/lang/String;)Landroidx/core/content/FileProvider$PathStrategy; (0xcc) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Landroidx/core/os/EnvironmentCompat;->getStorageState(Ljava/io/File;)Ljava/lang/String; (0x34) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lch/qos/logback/core/android/AndroidContextUtil;->getExternalStorageDirectoryPath()Ljava/lang/String; (0x0) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lch/qos/logback/core/android/AndroidContextUtil;->getMountedExternalStorageDirectoryPath()Ljava/lang/String; (0x2e) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
```



```
=>Lcom/stryker/settings/utils/Utils$Companion;->getFilePath(Landroid/content/Context;)Ljava/lang/String; (0x22) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=> Lcom/stryker/settings/utils/Utils$Companion;->appendLog(Ljava/lang/String;)V (0x12) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=> Lcom/stryker/settings/utils/Utils$Companion;->appendLog(Ljava/lang/String;)V (0x4a) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lcom/stryker/softwareupgrade/utils/UploadLogsController;-
    >downloadSwApkFromBlob(Lcom/stryker/softwareupgrade/utils/Constan
        ts$FMFileDownloadListener; Z Ljava/lang/String; Ljava/lang/String; Z Ljava/lang/String; Z)V (0x40) --->
        Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lcom/stryker/softwareupgrade/utils/UploadLogsController;-
    >swApkFromBlobFileSize(Lcom/stryker/softwareupgrade/utils/Constan
        ts$FMFileDownloadListener; Z Ljava/lang/String; Ljava/lang/String;)V (0x22) --->
        Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lcom/stryker/softwareupgrade/services/SoftwareUpgradeService;->callbackIndividualFirmwareUpdate(Ljava/lang/Boolean;
    Ljava/lang/String; Ljava/lang/Boolean; Ljava/lang/String; Z)V (0xbe) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=> Lcom/stryker/softwareupgrade/ui/SoftwareUpgradeActivity;->callback(Ljava/lang/Boolean; Ljava/lang/String;
    Ljava/lang/Boolean;)V (0x66) ---> Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
=>Lcom/stryker/softwareupgrade/ui/SoftwareUpgradeActivity;->callbackIndividualFirmwareUpdate(Ljava/lang/Boolean;
    Ljava/lang/String; Ljava/lang/Boolean; Ljava/lang/String; Z)V (0x24) --->
    Landroid/os/Environment;->getExternalStorageDirectory()Ljava/io/File;
```

AndroidManifest Exported Components Checking:

Found "exported" components(except for Launcher) for receiving outside applications' actions (AndroidManifest.xml).

These components can be initialized by other apps. You should add or modify the attribute to [exported="false"] if you don't want to.

You can also protect it with a customized permission with "signature" or higher protection Level and specify in "android:permission" attribute.

```
service => com.stryker.softwareupgrade.services.SoftwareUpgradeService
```

<Sensitive_Information> Getting IMEI and Device ID:

This app has code getting the "device id(IMEI)" but there are problems with this "TelephonyManager.getDeviceId()" approach.

1.Non-phones: Wifi-only devices or music players that don't have telephony hardware just don't have this kind of unique identifier.

2.Persistence: On devices which do have this, it persists across device data wipes and factory resets. It's not clear at all if, in this situation, your app should regard this as the same device.

3.Privilege:It requires READ_PHONE_STATE permission, which is irritating if you don't otherwise use or need telephony.



4.Bugs: We have seen a few instances of production phones for which the implementation is buggy and returns garbage, for example zeros or asterisks.

If you want to get an unique id for the device, we suggest you use "Installation" framework in the following article.

Please check the reference: <http://android-developers.blogspot.tw/2011/03/identifying-app-installations.html>

```
=>Landroidx/core/telephony/TelephonyManagerCompat;->getImei(Landroid/telephony/TelephonyManager;)Ljava/lang/String;  
(0xb0)  
---> Landroid/telephony/TelephonyManager;->getDeviceId()Ljava/lang/String;
```

CONFIDENTIAL



Accurate **Detection**.
Real Time **Response**.
Faster **Recovery**.



your business,
your brand, and
everything in between

www.gsecurelabs.com

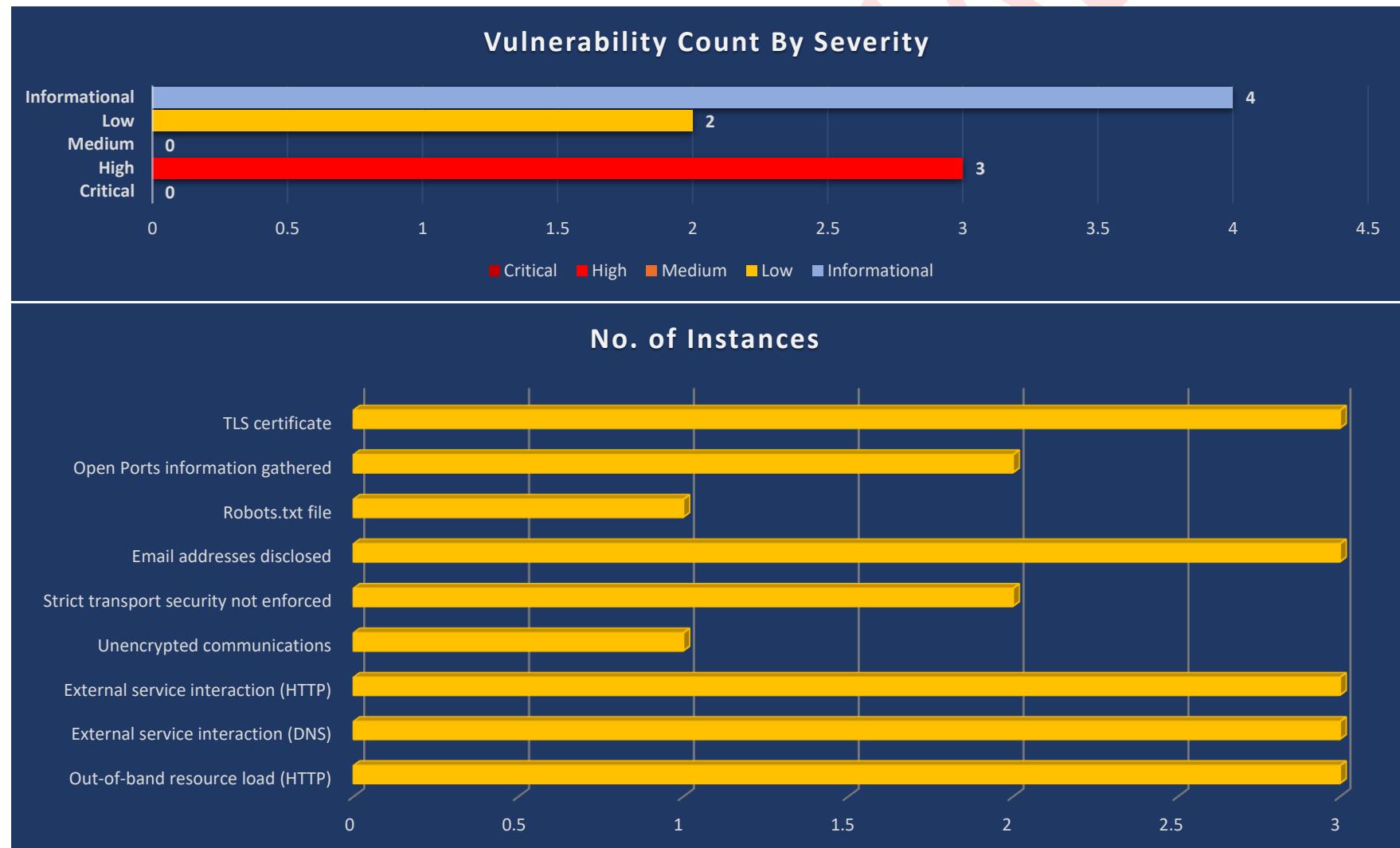
Nurse Station and Smart Medic Azure Applications



5. Technical Report of “Nurse Station and Smart Medic Azure Application” Project Vulnerability Assessment

The section uncovers the security checkups, which presents the findings of a security assessment conducted in our web applications and APIs.

This section reveals the exact areas where your Smart Medic Solution project could be exposed to security threats, areas needed to be improved and it will recommend the solutions to address such risks.



5.1 Out-of-band resource load (HTTP)

There are 3 instances of this issue:

- <https://smartmedic-testing.com>
- <https://admin.smartmedic-testing.com>
- <https://api.smartmedic-testing.com/>

Issue background:

Out-of-band resource load arises when it is possible to induce an application to fetch content from an arbitrary external location, and incorporate that content into the application's own response(s). The ability to trigger arbitrary out-of-band resource load does not constitute a vulnerability in its own right, and in some cases might even be the intended behavior of the application. However, in many cases, it can indicate a vulnerability with serious consequences.

The ability to request and retrieve web content from other systems can allow the application server to be used as a two-way attack proxy. By submitting suitable payloads, an attacker can cause the application server to attack, or retrieve content from, other systems that it can interact with. This may include public third-party systems, internal systems within the same organization, or services available on the local loopback adapter of the application server itself. Depending on the network architecture, this may expose highly vulnerable internal services that are not otherwise accessible to external attackers.

Additionally, the application's processing of web content that is retrieved from arbitrary URLs exposes some important and non-conventional attack surface. An attacker can deploy a web server that returns malicious content, and then induce the application to retrieve and process that content. This processing might give rise to the types of input-based vulnerabilities that are normally found when unexpected input is submitted directly in requests to the application. The out-of-band attack surface that the application exposes should be thoroughly tested for these types of vulnerabilities.

Issue remediation:

You should review the purpose and intended use of the relevant application functionality, and determine whether the ability to trigger arbitrary out-of-band resource load is intended behavior. If so, you should be aware of the types of attacks that can be performed via this behavior and take appropriate measures. These measures might include blocking network access from the application server to other internal systems, and hardening the application server itself to remove any services available on the local loopback adapter. You should also ensure that content retrieved from other systems is processed in a safe manner, with the usual precautions that are applicable when processing input from direct incoming web requests.

If the ability to trigger arbitrary out-of-band resource load is not intended behavior, then you should implement a whitelist of permitted URLs, and block requests to URLs that do not appear on this whitelist.



References:

- [Burp Collaborator](#)
- [Out-of-band application security testing \(OAST\)](#)
- [PortSwigger Research: Cracking the Lens](#)

Vulnerability classifications:

- [CWE-610: Externally Controlled Reference to a Resource in Another Sphere](#)
- [CWE-918: Server-Side Request Forgery \(SSRF\)](#)

1. <https://smartmedic-testing.com>

Severity: High

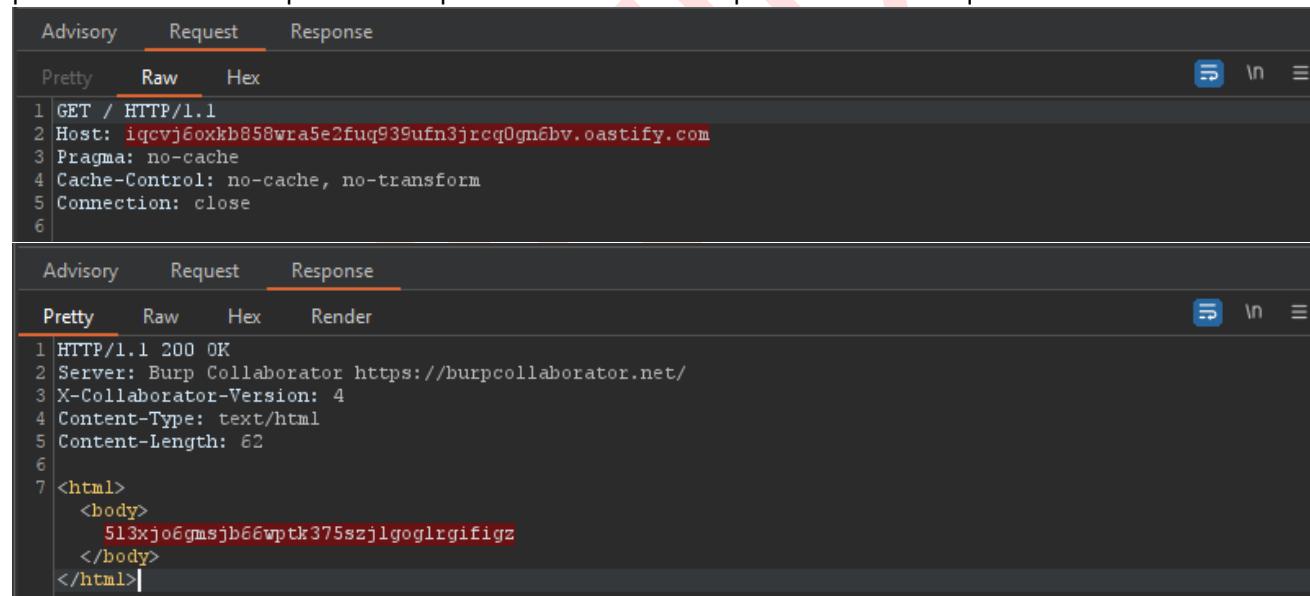
Confidence: Certain

Host: https://smartmedic-testing.com

Path: /

Issue detail:

It is possible to induce the application to retrieve the contents of an arbitrary external URL and return those contents in its own response. The payload **iqcvj6oxkb858wra5e2fuq939ufn3jrcq0gn6bv.oastify.com** was submitted in the SSL SNI value and the HTTP Host header. The application performed an HTTP request to the specified domain. The response from that request was then included in the application's own response.



The screenshot shows two panels of the Burp Suite interface. The top panel displays an OAST test configuration with tabs for 'Advisory', 'Request', and 'Response'. The 'Request' tab is selected, showing a raw HTTP GET request to the host 'iqcvj6oxkb858wra5e2fuq939ufn3jrcq0gn6bv.oastify.com'. The bottom panel shows the resulting response, which includes an HTTP header and the body of the external page, indicating successful SSRF execution.

Line	Content
1	GET / HTTP/1.1
2	Host: iqcvj6oxkb858wra5e2fuq939ufn3jrcq0gn6bv.oastify.com
3	Pragma: no-cache
4	Cache-Control: no-cache, no-transform
5	Connection: close
6	

Line	Content
1	HTTP/1.1 200 OK
2	Server: Burp Collaborator https://burpcollaborator.net/
3	X-Collaborator-Version: 4
4	Content-Type: text/html
5	Content-Length: 62
6	
7	<html>
	<body>
	513xjo6gmsjb66wptk375szjlgoqlrgifigz
	</body>
	</html>



2. <https://api.smartmedic-testing.com/api/admin/getwebapplatestversion>

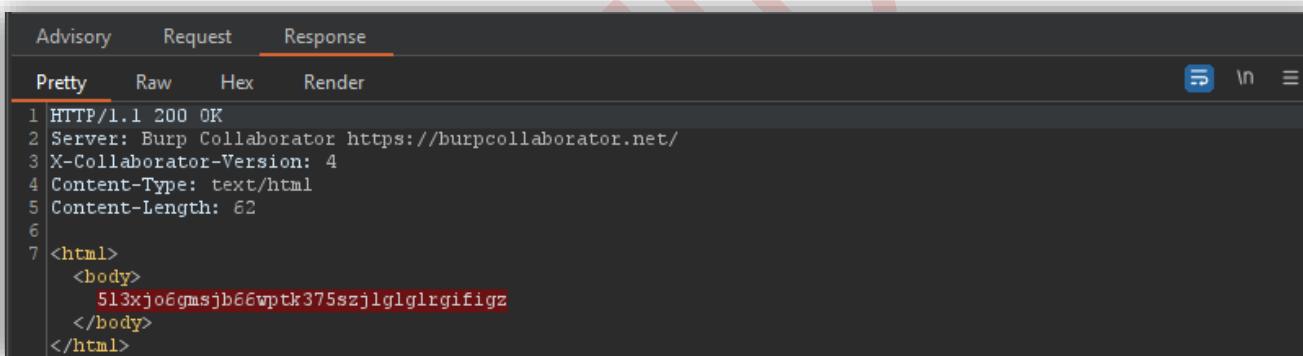
Severity: High
Confidence: Certain
Host: https://api.smartmedic-testing.com
Path: /api/admin/getwebapplatestversion

Issue detail:

It is possible to induce the application to retrieve the contents of an arbitrary external URL and return those contents in its own response. The payload **xopahlmc iq6k6bpp3t0us57i79d21vpoocez4nt.oastify.com** was submitted in the SSL SNI value and the HTTP Host header. The application performed an HTTP request to the specified domain. The response from that request was then included in the application's own response.



```
Advisory Request Response
Pretty Raw Hex
1 GET / HTTP/1.1
2 Host: xopahlmc iq6k6bpp3t0us57i79d21vpoocez4nt.oastify.com
3 Pragma: no-cache
4 Cache-Control: no-cache, no-transform
5 Connection: close
```



```
Advisory Request Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
8   <body>
9     513xj06gmsjb66wptk375szjlglrlrgifigz
10   </body>
11 </html>
```

3. <https://admin.smartmedic-testing.com>

Severity: High
Confidence: Certain



Host: https://admin.smartmedic-testing.com/
Path: /

Issue detail:

It is possible to induce the application to retrieve the contents of an arbitrary external URL and return those contents in its own response. The payload **bo3ohzmqi46y6pp33708sj7w7ndg1ep7ovei46t.oastify.com** was submitted in the SSL SNI value and the HTTP Host header. The application performed an HTTPS request to the specified domain. The response from that request was then included in the application's own response.

Advisory Request Response Collaborator HTTP interaction

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: bo3ohzmqi46y6pp33708sj7w7ndg1ep7ovei46t.oastify.com
3 Pragma: no-cache
4 Cache-Control: no-cache, no-transform
5 Connection: close
6
```

Advisory Request Response Collaborator HTTP interaction

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
8   <body>
9     513xjo6gmsjb66wptk375szjlgqqlrgifigz
10   </body>
11 </html>
```

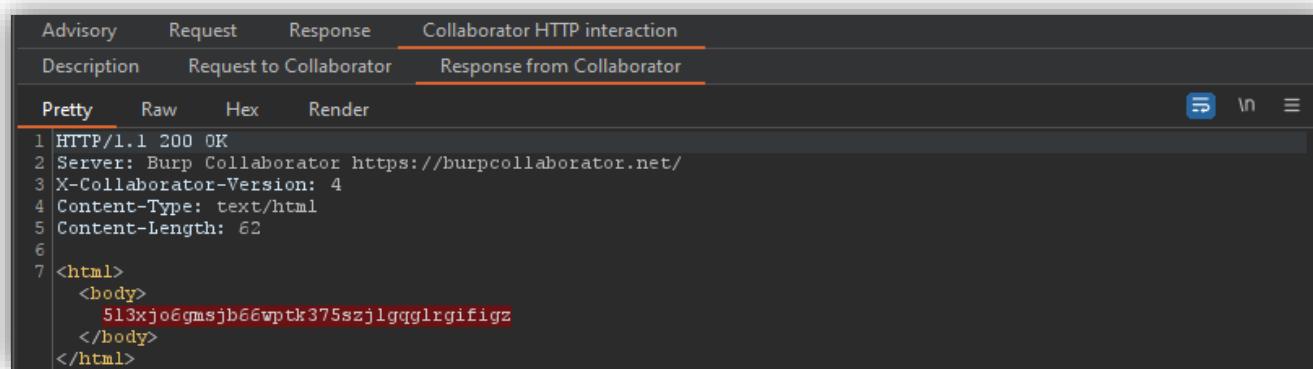
Advisory Request Response Collaborator HTTP interaction

Description Request to Collaborator Response from Collaborator

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: bo3ohzmqi46y6pp33708sj7w7ndg1ep7ovei46t.oastify.com
3 Pragma: no-cache
4 Cache-Control: no-cache, no-transform
5 Connection: close
```





```
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
<body>
513xjo6gmsjb66wptk375szjlguglrgifigz
</body>
</html>
```

5.2 External service interaction (DNS)

There are 3 instances of this issue:

- <https://smartmedic-testing.com>
- <https://api.smartmedic-testing.com>
- <https://admin.smartmedic-testing.com>

Issue background:

External service interaction arises when it is possible to induce an application to interact with an arbitrary external service, such as a web or mail server. The ability to trigger arbitrary external service interactions does not constitute a vulnerability in its own right, and in some cases might even be the intended behavior of the application. However, in many cases, it can indicate a vulnerability with serious consequences.

In cases where DNS-based interactions can be triggered, it is normally possible to trigger interactions using other service types, and these are reported as separate issues. If a payload that specifies a particular service type (e.g. a URL) triggers only a DNS-based interaction, then this strongly indicates that the application attempted to connect using that other service, but was prevented from doing so by egress filters in place at the network layer. The ability to send requests to other systems can allow the vulnerable server to be used as an attack proxy. By submitting suitable payloads, an attacker can cause the application server to attack other systems that it can interact with. This may include public third-party systems, internal systems within the same organization, or services available on the local loopback adapter of the application server itself. Depending on the network architecture, this may expose highly vulnerable internal services that are not otherwise accessible to external attackers.

Issue remediation:



You should review the purpose and intended use of the relevant application functionality, and determine whether the ability to trigger arbitrary external service interactions is intended behavior. If so, you should be aware of the types of attacks that can be performed via this behavior and take appropriate measures. These measures might include blocking network access from the application server to other internal systems, and hardening the application server itself to remove any services available on the local loopback adapter.

If the ability to trigger arbitrary external service interactions is not intended behavior, then you should implement a whitelist of permitted services and hosts, and block any interactions that do not appear on this whitelist.

Out-of-Band Application Security Testing (OAST) is highly effective at uncovering high-risk features, to the point where finding the root cause of an interaction can be quite challenging. To find the source of an external service interaction, try to identify whether it is triggered by specific application functionality, or occurs indiscriminately on all requests. If it occurs on all endpoints, a front-end CDN or application firewall may be responsible, or a back-end analytics system parsing server logs. In some cases, interactions may originate from third-party systems; for example, a HTTP request may trigger a poisoned email which passes through a link-scanner on its way to the recipient.

References:

- [Burp Collaborator](#)
- [Out-of-band application security testing \(OAST\)](#)
- [PortSwigger Research: Cracking the Lens](#)

Vulnerability classifications:

- [CWE-610: Externally Controlled Reference to a Resource in Another Sphere](#)
- [CWE-918: Server-Side Request Forgery \(SSRF\)](#)
- [CWE-406: Insufficient Control of Network Message Volume \(Network Amplification\)](#)

1. <https://smartmedic-testing.com>

Severity: High
Confidence: Certain
Host: <https://smartmedic-testing.com>
Path: /

Issue detail:

It is possible to induce the application to perform server-side DNS lookups of arbitrary domain names. The payload **dubqn1sso6c0crv5996ayldydpji7ev7uvkia6z.oastify.com** was submitted in the SSL SNI value and the HTTP Host header. The application performed a DNS lookup of the specified domain.



	Advisory	Request	Response	Collaborator DNS interaction
	Pretty	Raw	Hex	
1	GET / HTTP/1.1			
2	Host: dubqnlss0fc0crv5996ayldydpji7ev7uvkia6z.oastify.com			
3	Pragma: no-cache			
4	Cache-Control: no-cache, no-transform			
5	Connection: close			

Advisory Request Response Collaborator DNS interaction

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
8   <body>
9     613xjo6gmsjb66wptk375szjlgoglrgifigz
10    </body>
11  </html>
```

2. <https://api.smartmedic-testing.com/api/admin/getwebappplatestversion>

Severity: High
Confidence: Certain



Host: https://api.smartmedic-testing.com
Path: /api/admin/getwebappplatetestversion

Issue detail:

It is possible to induce the application to perform server-side DNS lookups of arbitrary domain names. The payload **hqbuj5owka848vr95d2eup929tfm3fr8qwgj67v.oastify.com** was submitted in the SSL SNI value and the HTTP Host header. The application performed a DNS lookup of the specified domain.

Advisory Request Response Collaborator DNS interaction

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: hqbuj5owka848vr95d2eup929tfm3fr8qwgj67v.oastify.com
3 Pragma: no-cache
4 Cache-Control: no-cache, no-transform
5 Connection: close
6
```

Advisory Request Response Collaborator DNS interaction

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
8   <body>
9     513xjo6gmsjb66wptk375szjlglrgifigz
10    </body>
11  </html>
```

Advisory	Request	Response	Collaborator DNS interaction
Description	DNS query		
Hex	Raw		
00000000	6f 63 00 00 00 01 00 00 00 00 00 00 00 27 68 71 62	oc\b'hqb	
00000010	75 6a 35 6f 77 6b 61 38 34 38 76 72 39 35 64 32	uj5owka848vr95d2	
00000020	65 75 70 39 32 39 74 66 6d 33 66 72 38 71 77 67	eup929tfm3fr8qwg	
00000030	6a 36 37 76 07 6f 61 73 74 69 66 79 03 63 6f 6d	j67v\oastify\com	
00000040	00 00 01 00 01 -- -- -- -- -- -- -- -- -- -- --	\00	



3. <https://admin.smartmedic-testing.com>

Severity: High
Confidence: Certain
Host: https://admin.smartmedic-testing.com
Path: /

Issue detail:

Issue detail

It is possible to induce the application to perform server-side DNS lookups of arbitrary domain names. The payload **8w5lpwunq1evemx0b4850gftfkld9bx4wsmfc31.oastify.com** was submitted in the SSL SNI value and the HTTP Host header. The application performed a DNS lookup of the specified domain.



5.3 External service interaction (HTTP)

There are 3 instances of this issue:

- <https://smartmedic-testing.com>
- <https://api.smartmedic-testing.com>
- <https://admin.smartmedic-testing.com>

Issue background:

External service interaction arises when it is possible to induce an application to interact with an arbitrary external service, such as a web or mail server. The ability to trigger arbitrary external service interactions does not constitute a vulnerability in its own right, and in some cases might even be the intended behavior of the application. However, in many cases, it can indicate a vulnerability with serious consequences.

The ability to send requests to other systems can allow the vulnerable server to be used as an attack proxy. By submitting suitable payloads, an attacker can cause the application server to attack other systems that it can interact with. This may include public third-party systems, internal systems within the same organization, or services available on the local loopback adapter of the application server itself. Depending on the network architecture, this may expose highly vulnerable internal services that are not otherwise accessible to external attackers.

Issue remediation:

You should review the purpose and intended use of the relevant application functionality, and determine whether the ability to trigger arbitrary external service interactions is intended behavior. If so, you should be aware of the types of attacks that can be performed via this behavior and take appropriate measures. These measures might include blocking network access from the application server to other internal systems, and hardening the application server itself to remove any services available on the local loopback adapter.

If the ability to trigger arbitrary external service interactions is not intended behavior, then you should implement a whitelist of permitted services and hosts, and block any interactions that do not appear on this whitelist.

Out-of-Band Application Security Testing (OAST) is highly effective at uncovering high-risk features, to the point where finding the root cause of an interaction can be quite challenging. To find the source of an external service interaction, try to identify whether it is triggered by specific application functionality, or occurs indiscriminately on all requests. If it occurs on all endpoints, a front-end CDN or application firewall may be responsible, or a back-end analytics system parsing server logs. In some cases, interactions may originate from third-party systems; for example, a HTTP request may trigger a poisoned email which passes through a link-scanner on its way to the recipient.

References:

- [Burp Collaborator](#)
- [Out-of-band application security testing \(OAST\)](#)
- [PortSwigger Research: Cracking the Lens](#)



Vulnerability classifications:

- [CWE-918: Server-Side Request Forgery \(SSRF\)](#)
- [CWE-406: Insufficient Control of Network Message Volume \(Network Amplification\)](#)

1. <https://smartmedic-testing.com>

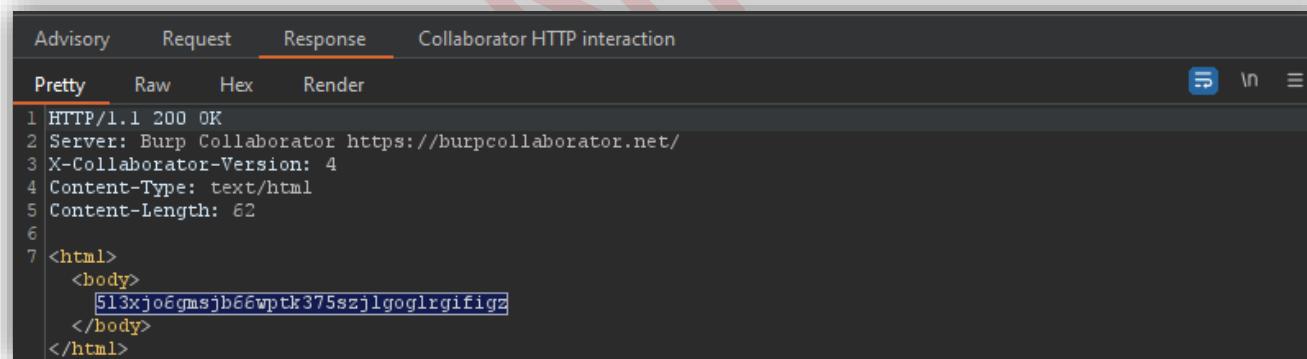
Severity: High
Confidence: Certain
Host: https://smartmedic-testing.com
Path: /

Issue detail:

It is possible to induce the application to perform server-side HTTPS requests to arbitrary domains. The payload **1ace3p8g4usosfbtpxmye9tmtdz6n2bvaj06quf.oastify.com** was submitted in the SSL SNI value and the HTTP Host header. The application performed an HTTPS request to the specified domain.



```
1 GET / HTTP/1.1
2 Host: 1ace3p8g4usosfbtpxmye9tmtdz6n2bvaj06quf.oastify.com
3 Pragma: no-cache
4 Cache-Control: no-cache, no-transform
5 Connection: close
```



```
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
8   <body>
9     513xjo6gmsjb66wptk375szjlgoglrgifigz
10   </body>
11 </html>
```



Advisory Request Response Collaborator HTTP interaction

Description Request to Collaborator Response from Collaborator

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: lace3p8g4usosfbtpxmye9tmtdz6n2bvaj06quf.oastify.com
3 Pragma: no-cache
4 Cache-Control: no-cache, no-transform
5 Connection: close
```

Advisory Request Response Collaborator HTTP interaction

Description Request to Collaborator Response from Collaborator

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
8   <body>
9     513xjo6gmsjb66wptk375szj1goglrgifigz
10   </body>
11 </html>
```

2. <https://admin.smartmedic-testing.com>

Severity: High
Confidence: Certain
Host: <https://admin.smartmedic-testing.com>
Path: /

Issue detail:

It is possible to induce the application to perform server-side HTTPS requests to arbitrary domains. The payload **usq7liq9mnaha8tm7q4rw2fbfb6hz5xtqsei18px.oastify.com** was submitted in the SSL SNI value and the HTTP Host header. The application performed an HTTPS request to the specified domain.



Advisory Request Response Collaborator HTTP interaction

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: usq7liq9mnaha8tm7q4rw2bfb6hz5xtqseil8px.oastify.com
3 Pragma: no-cache
4 Cache-Control: no-cache, no-transform
5 Connection: close
6
```

Advisory Request Response Collaborator HTTP interaction

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
8   <body>
9     513xjo6gmsjb66wptk375szjlgqqlrgifigz
10    </body>
11  </html>
```

Advisory Request Response Collaborator HTTP interaction

Description Request to Collaborator Response from Collaborator

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: usq7liq9mnaha8tm7q4rw2bfb6hz5xtqseil8px.oastify.com
3 Pragma: no-cache
4 Cache-Control: no-cache, no-transform
5 Connection: close
6
```



Advisory	Request	Response	Collaborator HTTP interaction
Description	Request to Collaborator	Response from Collaborator	
			<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: Burp Collaborator https://burpcollaborator.net/ 3 X-Collaborator-Version: 4 4 Content-Type: text/html 5 Content-Length: 62 6 7 <html> 8 <body> 9 513xj06gmsjb66wptk375szj1ggqlrgifigz 10 </body> 11 </html></pre>

3. <https://api.smartmedic-testing.com/api/admin/getwebappatestversion>

Severity: High
Confidence: Certain
Host: <https://api.smartmedic-testing.com>
Path: /api/admin/getwebappatestversion

Issue detail:

It is possible to induce the application to perform server-side HTTPS requests to arbitrary domains. The payload **dctq51as66u0urd5r9oaglvyp1ipbd4cs2fs3h.oastify.com** was submitted in the SSL SNI value and the HTTP Host header. The application performed an HTTPS request to the specified domain.

Advisory	Request	Response	Collaborator HTTP interaction
Description	Request to Collaborator	Response from Collaborator	
			<pre>Pretty Raw Hex 1 GET / HTTP/1.1 2 Host: dctq51as66u0urd5r9oaglvyp1ipbd4cs2fs3h.oastify.com 3 Pragma: no-cache 4 Cache-Control: no-cache, no-transform 5 Connection: close</pre>



Advisory Request Response Collaborator HTTP interaction

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
8   <body>
9     513xjo6gmsjb66wptk375szj1glglrgifigz
10    </body>
11 </html>
```

Advisory Request Response Collaborator HTTP interaction

Description Request to Collaborator Response from Collaborator

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: dctq5las66u0urd5r9oaglvyyplipbd4cs2fs3h.oastify.com
3 Pragma: no-cache
4 Cache-Control: no-cache, no-transform
5 Connection: close
```

Advisory Request Response Collaborator HTTP interaction

Description Request to Collaborator Response from Collaborator

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: Burp Collaborator https://burpcollaborator.net/
3 X-Collaborator-Version: 4
4 Content-Type: text/html
5 Content-Length: 62
6
7 <html>
8   <body>
9     513xjo6gmsjb66wptk375szj1glglrgifigz
10    </body>
11 </html>
```



5.4 Unencrypted communications

Severity: Low

Confidence: Certain

Host: http://smartmedic-testing.com

Path: /

Issue description

The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites. Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.

To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

Please note that using a mixture of encrypted and unencrypted communications is an ineffective defense against active attackers, because they can easily remove references to encrypted resources when these references are transmitted over an unencrypted connection.

Issue remediation

Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection.

References

- [Marking HTTP as non-secure](#)
- [Configuring Server-Side SSL/TLS](#)
- [HTTP Strict Transport Security](#)

Vulnerability classifications

- [CWE-326: Inadequate Encryption Strength](#)
- [CAPEC-94: Man in the Middle Attack](#)
- [CAPEC-157: Sniffing Attacks](#)



5.5 Strict transport security not enforced

There are 2 instances of this issue:

- <https://admin.smartmedic-testing.com/robots.txt>
- <https://api.smartmedic-testing.com/api>

Issue description

The application fails to prevent users from connecting to it over unencrypted connections. An attacker able to modify a legitimate user's network traffic could bypass the application's use of SSL/TLS encryption, and use the application as a platform for attacks against its users. This attack is performed by rewriting HTTPS links as HTTP, so that if a targeted user follows a link to the site from an HTTP page, their browser never attempts to use an encrypted connection. The `sslstrip` tool automates this process.

To exploit this vulnerability, an attacker must be suitably positioned to intercept and modify the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

Issue remediation

The application should instruct web browsers to only access the application using HTTPS. To do this, enable HTTP Strict Transport Security (HSTS) by adding a response header with the name 'Strict-Transport-Security' and the value 'max-age=expireTime', where expireTime is the time in seconds that browsers should remember that the site should only be accessed using HTTPS. Consider adding the 'includeSubDomains' flag if appropriate.

Note that because HSTS is a "trust on first use" (TOFU) protocol, a user who has never accessed the application will never have seen the HSTS header, and will therefore still be vulnerable to SSL stripping attacks. To mitigate this risk, you can optionally add the 'preload' flag to the HSTS header, and submit the domain for review by browser vendors.

References

- [HTTP Strict Transport Security](#)
- [sslstrip](#)
- [HSTS Preload Form](#)

Vulnerability classifications

- [CWE-523: Unprotected Transport of Credentials](#)
- [CAPEC-94: Man in the Middle Attack](#)
- [CAPEC-157: Sniffing Attacks](#)



1. <https://admin.smartmedic-testing.com/robots.txt>

Severity: Low

Confidence: Certain

Host: https://admin.smartmedic-testing.com

Path: /robots.txt

Request

```
GET /robots.txt HTTP/1.1
Host: admin.smartmedic-testing.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Upgrade-Insecure-Requests: 1
Accept-Encoding: gzip, deflate
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36
Connection: close
Cache-Control: max-age=0
```

Response

```
HTTP/1.1 200 OK
Content-Length: 67
Connection: close
Content-Type: text/plain
Date: Thu, 28 Jul 2022 13:29:40 GMT
Server: Microsoft-IIS/10.0
Accept-Ranges: bytes
Cache-Control: no-cache, no-store, must-revalidate, pre-check=0, post-check=0, max-age=0, s-maxage=0
ETag: "0ab48ab689bd81:0"
Expires: 0
Last-Modified: Tue, 19 Jul 2022 12:11:26 GMT
Pragma: no-cache
Set-Cookie:
ARRAffinity=7fe6542cfbcf1e24643dda274f773fec3e017e37888d5f3aa90706f6a0ce412c;Path=/;HttpOnly;Secure;Domain=admin.smartmedic-testing.com
```



```
Set-Cookie:  
ARRAffinitySameSite=7fe6542cfbcf1e24643dda274f773fec3e017e37888d5f3aa90706f6a0ce412c;Path=/;HttpOnly;SameSite=None;Secure;Domain=admin.smartmedic-testing.com  
Vary: Accept-Encoding  
X-Powered-By: ASP.NET  
X-Frame-Options: DENY  
  
# https://www.robotstxt.org/robotstxt.html  
User-agent: *  
Disallow:
```

2. https://api.smartmedic-testing.com/api

Severity: Low
Confidence: Certain
Host: https://api.smartmedic-testing.com
Path: /api

Request

```
GET /api HTTP/1.1  
Host: api.smartmedic-testing.com  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9  
Upgrade-Insecure-Requests: 1  
Accept-Encoding: gzip, deflate  
Accept-Language: en-US;q=0.9,en;q=0.8  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36  
Connection: close  
Cache-Control: max-age=0
```

Response

```
HTTP/1.1 401 Unauthorized  
Content-Length: 67  
Connection: close  
Content-Type: application/json; charset=utf-8  
Date: Thu, 28 Jul 2022 13:30:53 GMT
```



ETag: W/"43-aTCUkrxNXRDa0reW8Heu3O4IBk"

Vary: Origin

X-Powered-By: Express

X-Frame-Options: SAMEORIGIN

```
{"status":false,"statusCode":401,"data":{"message":"Unauthorize!"}}
```

5.6 Email addresses disclosed

There are 4 instances of this issue:

- <https://admin.smartmedic-testing.com/static/js/2.82452bfd.chunk.js>
- <https://admin.smartmedic-testing.com/static/js/main.103fa717.chunk.js>
- <https://smartmedic-testing.com/static/js/2.1a42a5cb.chunk.js>
- <https://smartmedic-testing.com/static/js/main.7853f144.chunk.js>

Issue background:

The presence of email addresses within application responses does not necessarily constitute a security vulnerability. Email addresses may appear intentionally within contact information, and many applications (such as web mail) include arbitrary third-party email addresses within their core content.

However, email addresses of developers and other individuals (whether appearing on-screen or hidden within page source) may disclose information that is useful to an attacker; for example, they may represent usernames that can be used at the application's login, and they may be used in social engineering attacks against the organization's personnel. Unnecessary or excessive disclosure of email addresses may also lead to an increase in the volume of spam email received.

Issue remediation

Consider removing any email addresses that are unnecessary, or replacing personal addresses with anonymous mailbox addresses (such as helpdesk@example.com).

To reduce the quantity of spam sent to anonymous mailbox addresses, consider hiding the email address and instead providing a form that generates the email server-side, protected by a CAPTCHA if necessary.

References:

- [Web Security Academy: Information disclosure](#)



Vulnerability classifications:

- [CWE-200: Information Exposure](#)
- [CAPEC-37: Retrieve Embedded Sensitive Data](#)

1. <https://admin.smartmedic-testing.com/static/js/2.82452bfd.chunk.js>

Severity: Informational

Confidence: Certain

Host: https://admin.smartmedic-testing.com

Path: /static/js/2.82452bfd.chunk.js

Issue detail:

The following email addresses were disclosed in the response:

- aes128-gcm@openssh.com
- aes192-gcm@openssh.com
- aes256-gcm@openssh.com
- git@github.com
- fedor@indutny.com
- -cert-v01@openssh.com
- ssh-rsa-cert-v01@openssh.com
- ssh-dss-cert-v01@openssh.com
- ssh-ed25519-cert-v01@openssh.com

Advisory	Request	Response
	Pretty	
	Hex	
1	GET /static/js/2.82452bfd.chunk.js HTTP/1.1	
2	Host: admin.smartmedic-testing.com	
3	Cookie: ARRAffinity=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546; ARRAffinitySameSite=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546	
4	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9	
5	Upgrade-Insecure-Requests: 1	
6	Accept-Encoding: gzip, deflate	
7	Accept-Language: en-US;q=0.9,en;q=0.8	
8	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36	
9	Connection: close	
10	Cache-Control: max-age=0	



```
HTTP/1.1 200 OK
Content-Length: 4550046
Connection: close
Content-Type: application/x-javascript
Date: Mon, 20 Jun 2022 13:31:20 GMT
Server: Microsoft-IIS/10.0
Accept-Ranges: bytes
Cache-Control: no-cache, no-store, must-revalidate, pre-check=0, post-check=0, max-age=0, s-maxage=0
ETag: "0d8f3ecef81d81:0"
Expires: 0
Last-Modified: Fri, 17 Jun 2022 02:14:08 GMT
Pragma: no-cache
Set-Cookie: ARRAffinity=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546;Path=/;HttpOnly;Secure;Domain=admin.smartmedic-testing.com
Set-Cookie: ARRAffinitySameSite=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546;Path=/;HttpOnly;SameSite=None;Secure;Domain=admin.smartmedic-testing.com
Vary: Accept-Encoding
X-Powered-By: ASP.NET

/*! For license information please see 2.82452bfd.chunk.js.LICENSE.txt */
(this["webpackJsonpdemo-nsa-admin-app"]的文化["webpackJsonpdemo-nsa-admin-app"] || []).push([[[2], [function(e, t, r) {"use strict"; e.
...[SNIP]...
switch(e){case "3des-cbc":t.keySize=24,t.blockSize=8,t.opensslName="des-ede3-cbc";break;case "blowfish-cbc":t.keySize=16,t.blockSize=8,t.opensslName="bf-cbc";break;case "aes128-cbc":case "aes128-ctr":case "aes128-gcm@openssh.com":t.keySize=16,t.blockSize=16,t.opensslName="aes-128-"+e.slice(7,10);break;case "aes192-cbc":case "aes192-ctr":case "aes192-gcm@openssh.com":t.keySize=24,t.blockSize=16,t.opensslName="aes-192-"+e.slice(7,10);break;case "aes256-cbc":case "aes256-ctr":case "aes256-gcm@openssh.com":t.keySize=32,t.blockSize=16,t.opensslName="aes-256-"+e.slice(7,10);break;default:throw new Error("Unsupported openssl cipher '"+e+"")}return
t}.publicFromPrivateECDSA:function(e,t){n.string(e,"curve"
...[SNIP]...
ix":"npm run lint -- --fix","unit":"istanbul test _mocha --reporter=spec test/index.js","test":"npm run lint && npm run unit","version":"grunt dist && git add
dist/","repository":{"type":"git","url":"git@github.com:indutny/elliptic"},"keywords":["EC","Elliptic","curve","Cryptography"],"author":"Fedor Indutny <fedor@indutny.com>
...[SNIP]...
(30),i=r(110),a=r(43),o=r(34).Buffer,s=r(52),l=r(47),c=(r(48),r(130)),u=r(86),h=r(74),f=r(49),d=r(129);var p={user:1,host:2};Object.keys(p).forEach((function(e){p[p[e]]=e}));var m=/^ecdsa-sha2-[^@-]+)-cert-v01@openssh.com$/;function g(e,t,r){var a=new i({buffer:e}),o=a.readString();if(void 0!==t&&o!==t)throw new Error("SSH certificate algorithm mismatch");void 0==!=t&&(t=o);var
g={signatures:{}},g.signatures.openssh={},...[SNIP]...
Buffer("ssh")).l.toBuffer()}}function w(e){if("rsa"==!=e.type) return "ssh-rsa-cert-v01@openssh.com";if("dsa"==!=e.type) return "ssh-dss-cert-v01@openssh.com";if("ecdsa"==!=e.type) return "ecdsa-sha2-+e.curve+-cert-v01@openssh.com";if("ed25519"==!=e.type) return "ssh-ed25519-cert-v01@openssh.com";throw new Error("Unsupported key type "+e.type)}},function(e,t,r){var
n=r(314);e.exports={read:function(e,t){"string"!=!=typeof e&&(i.buffer(e,"buf"),e=e.toString("ascii"));var r,o,s=e.trim().split(/[
...[SNIP]...
```

2. <https://admin.smartmedic-testing.com/static/js/main.103fa717.chunk.js>

Severity: Informational
Confidence: Certain
Host: <https://admin.smartmedic-testing.com>
Path: /static/js/main.103fa717.chunk.js

Issue detail: The following email addresses were disclosed in the response:



- abc@xyz.com

- abc@stryker.com

- email@stryker.com

Advisory Request Response

Pretty Raw Hex

```

1 GET /static/js/main.103fa717.chunk.js HTTP/1.1
2 Host: admin.smartmedic-testing.com
3 Cookie: ARRAffinity=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546; ARRAffinitySameSite=
c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546
4 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.9
5 Upgrade-Insecure-Requests: 1
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US;q=0.9,en;q=0.8
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/102.0.5005.63 Safari/537.36
9 Connection: close
10 Cache-Control: max-age=0

```

Advisory Request Response

Pretty Raw Hex Render

```

firstName)&&(a="Please enter first name"),""==o.value&&(l="Please select role"),""==(null==N||void 0==N?
void 0:N.emailID)||I.isValidEmail||(e="Enter email id in format : abc@stryker.com"),(null==N||void 0==N?
void 0:N.phoneNumber.length)>3&&l===(null==N||void 0==N?void 0:N.phoneNumber.length)&&(i=
"Please enter 10 digit"),""==e&"==a&"==l&"==i){var n=(data:{request:(emailId:null==N||void 0==N?
void 0:N.emailID,firstName:null==N||void 0==N?void 0:N.firstName,lastName:null==N||void 0==N?void 0:N.
lastName,contactNumber:null==N||void 0==N?void 0:N.phoneNumber,roleID:o.value,isNotifyReport:v.isActive:d?
:0)});null!=t&&void 0==t&t.id?m(t.id,n):else{var s=C(null==N||void 0==N?void 0:N.emailID);k(Object(
_e.a)(Object(_e.a)({},I),{},(emailID:e,isValidEmail:""!==(null==N||void 0==N?void 0:N.emailID)&&firstName
:a,isFirstName:""!==(null==N||void 0==N?void 0:N.firstName),Role:l,isRole:0!==(null==o||void 0==o?void 0:
o.value),phoneNumber:i,isPhoneNumber:!((null==N||void 0==N?void 0:N.phoneNumber.length)>3)||l3===(null==N
||void 0==N?void 0:N.phoneNumber.length))))},children:["Save"])}):Object(on.jsx)("div",{children:
Object(on.jsx)("button",{className:"btn-filled",onClick:function(){x(!0)},children:"Edit"}))),Object(on.
jsxs)("div",{className:w?"row mt-3":"row disabled mt-3",children:[Object(on.jsxs)("div",{
className:"col-4 mb-3",children:[Object(on.jsx)("label", {className:"form-label required-field", children:"Email ID"}),
Object(on.jsx)("input", {type:"text",placeholder:"email@stryker.com",value:N.emailID,name:"emailID",onChange:
function(e){return T(e)},className:"form-control w-100"}),Object(on.jsx)("div", {className:"invalid-feedback",
position-relative mb-2",children:Object(on.jsx)("div", {className:"invalid-feedback", children:""!==(null==I

```

Advisory Request Response

Pretty Raw Hex Render

```

login_box",children:[Object(on.jsx)("div", {className: "login_box", children:[Object(on.jsx)("div", {
className:"mb-3 position-relative",children:[Object(on.jsx)("label", {className:"form-label", children:"Email"})
, Object(on.jsx)("input", {type:"text",name:"loginId",placeholder:"abc@xyz.com",value:l,onChange:function(e){
return t.handleChange(e)},className:"form-control w-100"}),Object(on.jsx)("div", {className:"invalid-feedback"
,children:s.emailError}]}),Object(on.jsxs)("div", {className:"mb-5 position-relative",children:[Object(on.jsx)
("label", {className:"form-label", children:"Password"}),Object(on.jsx)("span", {className:"icon-eye",onClick:
this.togglePassword,children:n?Object(on.jsx)(vs.a,{}):Object(on.jsx)(js.a,{})}),Object(on.jsx)("input", {

```

3. <https://smartmedic-testing.com/static/js/2.1a42a5cb.chunk.js>

Severity: Informational

Confidence: Certain

Host: <https://smartmedic-testing.com>

Path: /static/js/2.1a42a5cb.chunk.js

Issue detail:

The following email addresses were disclosed in the response:

- git@github.com
- fedor@indutny.com



Advisory Request Response

Pretty Raw Hex

```
1 GET /static/js/2.1a42a5cb.chunk.js HTTP/1.1
2 Host: smartmedic-testing.com
3 Cookie: ARRAffinity=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546; ARRAffinitySameSite=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546;
mp_6221ebd1ebc258b3d56d226c287bc40b_mixpanel=%7B%22distinct_id%22%3A%20%22181814eda6050d-02ed64bcf8f66-4c2c7720-75300-181814eda61734%22%20%22%24device_id%22%3A%20%22181814eda6050d-02ed64bcf8f66-4c2c7720-75300-181814eda61734%22%20%22%24initial_referrer%22%3A%20%22%24direct%22%20%22%24initial_referring_domain%22%3A%20%22%24direct%22%7D
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
5 Upgrade-Insecure-Requests: 1
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US;q=0.9,en;q=0.8
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/102.0.5005.63 Safari/537.36
9 Connection: close
10 Cache-Control: max-age=0
```

Advisory Request Response

Pretty Raw Hex Render

```
,i,s,1),0==(v=i.invm(1).imul(g.add(o.mul(p))).mod(l)).cmpn(0)&&(v!=1,p=new a(0));return function(t,n){t=t.toArray(),n=n.toArray(),128&t[0]&&(t=[0].concat(t));128&n[0]&&(n=[0].concat(n));var r=[48,t.length+n.length+4,2,t.length];return r=r.concat(t,[2,n.length],n,new e(r))(p,v))(t,g,r)}if("rsa"!==d&&"ecdsa/rsa"!==d)throw new Error("wrong private key type");t=e.concat([p,t]);for(var v=g.modulus.byteLength(),m=[0,l];t.length+m.length+1<v;)m.push(255);m.push(0);for(var b=-1;+b<t.length;)m.push(t[b]);return i(m,g),t.exports.getKey=c,t.exports.makeKey=f}).call(this,n(30).Buffer)},function(t){t.exports=JSON.parse('{"name":"elliptic","version":"6.5.2","description":"EC cryptography","main":"lib/elliptic.js","files":["lib"],"scripts":{"jscs":"jscs benchmarks/*.js lib/*.js lib/**/*.*.js lib/test/index.js","jshint":"jscs benchmarks/*.js lib/*.js lib/**/*.*.js lib/**/*.*.js test/index.js","lint":"npm run jscs && npm run jshint","unit":"istanbul test _mocha --reporter spec test/index.js","test":"npm run lint && npm run unit","version":"grunt dist && git add dist/"},"repository":{"type":"git","url":"git@github.com:indutny/elliptic"},"keywords":["EC","Elliptic","curve","Cryptography"],"author":"Fedor Indutny <fedor@indutny.com>","license":"MIT","bugs":{"url":"https://github.com/indutny/elliptic/issues"},"homepage":"https://github.com/indutny/elliptic"}
```

4. <https://smartmedic-testing.com>

Severity: Informational
Confidence: Certain
Host: https://smartmedic-testing.com
Path: /static/js/main.7853f144.chunk.js

Issue detail:

The following email addresses were disclosed in the response:

- smartmedic@stryker.com
 - service.india@stryker.com
 - globalprivacy@stryker.com
 - apollo@gmail.com

Advisory Request Response

Pretty | Hex

1 GET /static/js/main.7853f144.chunk.js HTTP/1.1
2 Host: smartmedic-testing.com
3 Cookie: ARRAffinity=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546; ARRAffinitySameSite=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546;
mp_6221ebd16bc258b3d56d226c287bc40b_mixpanel=
%7B%2Bdistinct_id%22%3A%20%22181814ef4ab713-06df17462f5dc4-4c2c7720-75300-181814ef4accbb%22%2C%22%24device_id%22%3A%20%22181814ef4ab713-06df17462f5dc4-4c2c7720-75300-181814ef4accbb%22%2C%22%24initial_referrer%22%3A%20%22%24direct%22%2C%22%24initial_referring_domain%22%3A%20%22%24direct%22%7D
4 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
5 Upgrade-Insecure-Requests: 1
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US;q=0.9,en;q=0.8
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/102.0.5005.63 Safari/537.36
9 Connection: close
10 Cache-Control: max-age=0

Advisory Request Response

Pretty Raw Hex Render

```
"noshadow request-service-modal"],d.a.createElement("div",null,d.a.createElement("div",{"className: "text-center"},d.a.createElement("h2",null,"Customer Service"),d.a.createElement("p",null, "Please contact Stryker Customer Service"),d.a.createElement("p",{"className:"mb-20"},"Tel: ",d.a.createElement("b",null,"1800 103 8030"),d.a.createElement("br",null),"Email: ",d.a.createElement("a", {"className:"link-blue",href:"mailto:service.india@stryker.com"}, "service.india@stryker.com"))))),a(s, Component),Wa=function(e){Object(A.a)(a,e).var t=Object(p.a)(a);function a(e){var n;return Object(v.a)(this,a),n.t.call(this,e)).handleCardClick=function(e){var t;(O.a.a.track(null==Da)?void O==Da?void O:a=EventPadClick).Q---(null==e)?void Q---(e?void Q:a=connection status)|||null==e?void Q---(e?void Q:a=
```

Advisory Request Response

Pretty Raw Hex Render

```
You may always choose to object to the collection or use of your personal information or to request to have your information erased. If you would like a copy of the information held about you, or to exercise any other right, please contact us at ",d.a.createElement("a", {href:"mailto:globalprivacy@stryker.com"}, "globalprivacy@stryker.com"), ".),d.a.createElement("p",null, "If you like to change your settings for third party tracking cookies you can:"),d.a.createElement("ul",null, d.a.createElement("li",null,
```

Advisory Request Response

Pretty Raw Hex Render

```
na();,Karamzou, nt 4500,d.a.createElement("ul",null),d.a.createElement("p",null,d.a.createElement("a", {href:"mailto:globalprivacy@stryker.com"}, "globalprivacy@stryker.com")),d.a.createElement("p",null, "If we fail to respond to you within a reasonable period of receiving it in writing, or if you are dissatisfied with the response that you receive from us, you may lodge a complaint with the data protection authorities in your home country.")))]),a)(s.Component),yn=function(){return d.a.createElement("div", {className: "content-wrapper"},d.a.createElement(da,{isDashboardHeader:!0}),d.a.createElement("div", {className: "text-center mt-25"}),d.a.createElement("br",null),d.a.createElement("h2", {className: "title-30"}, "Customer Service"),d.a.createElement("p", {className: "title-24"}, "Please contact Stryker Customer Service"),d.a.createElement("p", {className: "title-20"}, "Tel: ",d.a.createElement("b",null,"1800 103 8030"),d.a.createElement("br",null),"Email: ",d.a.createElement("a", {className: "link-blue"}, href:"mailto:".concat("service.india@stryker.com"), "service.india@stryker.com")))),Cn=a(404),wn=a(181),Tn=function(e){Object(A.a)(a,e);var t=Object(p.a)(a);function a(){var e;return Object(v.a)(this,a),(e=t.call(this)).successFailureCallback=(SuccessCallback:function(t){var a,i=(null==t||void 0===t?void 0:t.data).data, result:!0===(null==t||void 0==t||null===(a=t.data)||void 0==a?void 0:a.status)?e.setState({allIframe:[t]}):e.setState({allIframe:[t]}))});},a(s.Component),jn=a(240),Vn=a.n(jn):a(745);var Pn=(getPatientReportURL:function(e,t){var a=t.SuccessCallback||FailureCallback||return console.log("getPatientReportURL: "+t);a(e,t)});var i=0,r=/mailto/;
```

Advisory Request Response

Pretty Raw Hex Render

```
You will be redirected to homepage in ,L, sec. .).d.a.createElement(da,{co: / }),on=function(e){Object(A.a)(a,e);var t=Object(p.a)(a);function a(){return Object(v.a)(this,a),t.apply(this,arguments)}return Object(m.a)(a,[{key:"render",value:function(){return d.a.createElement("div", {className: "content-wrapper"},d.a.createElement(va,null),d.a.createElement("div", {className: "text-center mt-25"}),d.a.createElement("br",null),d.a.createElement("h2", {className: "title-30"}, "Forgot Password?"),d.a.createElement("p", {className: "title-24"}, "Please contact Stryker Customer Service"),d.a.createElement("p", {className: "title-20"}, "Tel: ",d.a.createElement("b",null,"1800 103 8030"),d.a.createElement("br",null),"Email: ",d.a.createElement("a", {className: "link-blue"}, href:"mailto:".concat("service.india@stryker.com"), "service.india@stryker.com")))}],a)(s.Component),jn=a(240),Vn=a.n(jn):a(745);var Pn=(getPatientReportURL:function(e,t){var a=t.SuccessCallback||FailureCallback||return console.log("getPatientReportURL: "+t);a(e,t)});var i=0,r=/mailto/;
```

Advisory Request Response

Pretty Raw Hex Render

```
da, d.a.createElement("div",null,d.a.createElement("label", {className: "required-field"},this.props.t("SETTING.HOSPITALCODE"),{framework: "react-i18next"})),d.a.createElement("input", {maxLength:6,type:"text",value:i,ref:this.hospitalCode,onChange:this.handleHospitalCodeChange,placeholder:"12345"}),d.a.createElement("div", {className: "invalid-feedback"},this.getHospitalCodeError(1))),d.a.createElement("div", {className: "formController"},d.a.createElement("label", {className: "required-field"},this.props.t("SETTING.EMAIL"),{framework: "react-i18next"})),d.a.createElement("input", {type:"email",defaultValue:o,ref:this.email,onChange:this.handleEmailChange,placeholder:"apollo@gmail.com"}),d.a.createElement("div", {className: "invalid-feedback"},this.getHospitalEmailError(1))):d.a.createElement(Dn,{infoData:a))))}],a)(s.Component),Un=(updateHospitalInfo:function(e,t,a){var n=a.SuccessCallback,i=a.FailureCallback;return function(a){var l,r,s;a(ve()),a((type:null==o||void 0==o?void 0:qe)),ue("").concat(null==se||void 0==se||null==(l==se).UpdateHospitalInfo)||void 0==l?void 0:l.endpoint,"/").concat(e),t,null==se||void 0==se||null==(s==se.UpdateHospitalInfo)||void 0==s?void 0:s.method,(SuccessCallback:function(e){var t=(null==e||void 0==e?void 0:e.data).data,
```

```
Advisory Request Response
Pretty Raw Hex Render
=====(t=e.data)||void 0==t?void 0:t.status))if(null==e||void 0==e||null==(i=e.data)||void 0==i||  
null==(o=i.data)||void 0==o||null==(l=o.result)||void 0==l?void 0:l.hospitalId){var c,u,v=null==e||void  
0==e||null==(c=e.data)||void 0==c||null==(u=c.data)||void 0==u?void 0:u.result,m=null==v||void 0==v?  
void 0:v.hospitalCode,A=null==v||void 0==v?void 0:v.hospitalId,p=null==v||void 0==v?void 0:v.hospitalName  
,h=null==v||void 0==v?void 0:v.serviceRequestEmail):null==v||void 0==v?void 0:v.serviceRequestEmail:  
"smartmedic@stryker.com",E=(null==v||void 0==v?void 0:v.serviceRequestNumber)?null==v||void 0==v?void 0:v  
.serviceRequestNumber:"+91 931 959 4185",b=null==v||void 0==v?void 0:v.email:H(null==D||void 0==D?void 0:  
D.HOSPITALID,A),H(null==D||void 0==D?void 0:D.HOSPITALNAME,p),H(null==D||void 0==D?void 0:D.EMAIL,b),H(  
null==D||void 0==D?void 0:D.SERVICEREQUESTEMAIL,h),H(null==D||void 0==D?void 0:D.SERVICEREQUESTNUMBER,E),  
H(null==D||void 0==D?void 0:D.ISAUTENTICATED,JSON.stringify(!0));var g=(hospitalId:A,hospitalName:p,  
hospitalCode:m);n.props.localStorageValue(g),a(),n.props.history.replace("/dashboard")}else n.setState({
```

5.7 Robots.txt file

Severity: Information

Confidence: Certain

Host: https://admin.smartmedic-testing.com

Path: /robots.txt

Issue details:

The web server contains a robots.txt file.

Issue background:

The file robots.txt is used to give instructions to web robots, such as search engine crawlers, about locations within the web site that robots are allowed, or not allowed, to crawl and index.

The presence of the robots.txt does not in itself present any kind of security vulnerability. However, it is often used to identify restricted or private areas of a site's contents. The information in the file may therefore help an attacker to map out the site's contents, especially if some of the locations identified are not linked from elsewhere in the site. If the application relies on robots.txt to protect access to these areas, and does not enforce proper access control over them, then this presents a serious vulnerability.

Issue remediation

The robots.txt file is not itself a security threat, and its correct use can represent good practice for non-security reasons. You should not assume that all web robots will honor the file's instructions. Rather, assume that attackers will pay close attention to any locations identified in the file. Do not rely on robots.txt to provide any kind of protection over unauthorized access.

References:

- [Web Security Academy: Information disclosure](#)



Vulnerability classifications:

- [CWE-200: Information Exposure](#)

```
Advisory Request Response
Pretty Raw Hex
1 GET /robots.txt HTTP/1.1
2 Host: admin.smartmedic-testing.com
3 Accept-Encoding: gzip, deflate
4 Accept: */*
5 Accept-Language: en-US;q=0.9,en;q=0.8
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/102.0.5005.63 Safari/537.36
7 Connection: close
8 Cache-Control: max-age=0
9
```

```
Advisory Request Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Content-Length: 67
3 Connection: close
4 Content-Type: text/plain
5 Date: Mon, 20 Jun 2022 13:31:48 GMT
6 Server: Microsoft-IIS/10.0
7 Accept-Ranges: bytes
8 Cache-Control: no-cache, no-store, must-revalidate, pre-check=0, post-check=0, max-age=0, s-maxage=0
9 ETag: "0d30a9ef81d81:0"
10 Expires: 0
11 Last-Modified: Fri, 17 Jun 2022 02:12:14 GMT
12 Pragma: no-cache
13 Set-Cookie: APPAffinity=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546;
  Path=/;HttpOnly;Secure;Domain=admin.smartmedic-testing.com
14 Set-Cookie: APPAffinitySameSite=c02bd95f6de966f537132af58810d7960a1880e2261379fc4d49dd59e700e546;
  Path=/;HttpOnly;SameSite=None;Secure;Domain=admin.smartmedic-testing.com
15 Vary: Accept-Encoding
16 X-Powered-By: ASP.NET
17
18 # https://www.robotstxt.org/robotstxt.html
19 User-agent: *
20 Disallow:
21
```

5.8 Open Ports information gathered

Severity: Information

Confidence: Certain

Host: https://admin.smartmedic-testing.com

https://smartmedic-testing.com

Path: /

Issue details:

Open Ports information gathered,

Issue background:

Open ports become dangerous when legitimate services are exploited through security vulnerabilities or malicious services are introduced to a system via malware or social engineering, cybercriminals can use these services in conjunction with open ports to gain unauthorized access to sensitive data.

In security parlance, the term open port is used to mean a TCP or UDP port number that is configured to accept packets. In contrast, a port which rejects connections or ignores all packets directed at it is called a closed port.

To run an exploit, an attacker needs a vulnerability. To fingerprint a service, the attacker needs to know that there is one running on a publicly accessible port. To find out which publicly accessible ports run services, the attacker needs to run a port scan.

Although you do require ports to be open for users to connect to your services, you should restrict open ports, and ports exposed on the Internet to these services only.

Open ports allow attacker to:

- Configure the service to distribute content: Unused services tend to be left with default configurations, which are not always secure or may be using default passwords.
- Exploit old versions of unused software: Unused services tend to be forgotten, which means that they not get updated. Old versions of software tend to be full of known vulnerabilities.
- Gain better information on your network: Some services give an attacker easy access to certain information, at the very least, they can have a very good guess on the operating system that the server is running, which is already a good head start.

References:

- [Web Security Academy: Information disclosure](#)
- <https://docs.microsoft.com/en-us/azure/app-service/environment/app-service-app-service-environment-control-inbound-traffic>
- <https://social.msdn.microsoft.com/Forums/en-US/5a4c97bc-a3b5-456f-ae02-dfd7cdcc1caf/is-it-mandatory-to-close-1221-port?forum=azureappconfiguration>



Vulnerability classifications:

- [CWE-200: Information Exposure](#)

```
80/tcp open http?
443/tcp open ssl/https?
443/tcp open ssl/https Microsoft-IIS/10.0
454/tcp open ssl/upnp Microsoft IIS httpd
1221/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
4022/tcp open dnox?
4024/tcp open tnp1-port?
8172/tcp open ssl/http Microsoft IIS httpd 10.0
```

PORT STATE SERVICE VERSION

```
80/tcp open http
443/tcp open ssl/https Microsoft-IIS/10.0
454/tcp open ssl/upnp Microsoft IIS httpd
| ssl-cert: Subject: commonName=waws-prod-sg1-069.api.azurewebsites.windows.net/organizationName=Microsoft
Corporation/stateOrProvinceName=Washington/countryName=US
| Subject Alternative Name: DNS:waws-prod-sg1-069.api.azurewebsites.windows.net
| Issuer: commonName=DigiCert SHA2 Secure Server CA/organizationName=DigiCert Inc/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-07-02T00:00:00
| Not valid after: 2023-07-02T23:59:59
| MD5: f9b7 6b14 d410 9d51 943f d54d 3f0a 2cd9
| _SHA-1: a69f 7292 82b7 16c0 709b b27e bace 85d0 4984 1d92
| tls-alpn:
|   h2
|   http/1.1
| _ssl-date: 2022-08-02T16:21:46+00:00; 0s from scanner time.
1221/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| _http-favicon: Unknown favicon MD5: AD117D22DD5CA3E20E71E08A225E4121
| _http-title: Site doesn't have a title (text/plain).
```



```
|_http-trane-info: Problem with XML parsing of /evox/about
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Microsoft-HTTPAPI/2.0
4022/tcp open dnox?
4024/tcp open tnp1-port?
8172/tcp open ssl/http Microsoft IIS httpd 10.0
|_http-title: Site doesn't have a title (text/html).
| tls-alpn:
|_ h2
|_ http/1.1
| ssl-cert: Subject: commonName=waws-prod-sg1-069.publish.azurewebsites.windows.net/organizationName=Microsoft
Corporation/stateOrProvinceName=Washington/countryName=US
| Subject Alternative Name: DNS:waws-prod-sg1-069.publish.azurewebsites.windows.net, DNS:waws-prod-sg1-069.ftp.azurewebsites.windows.net
| Issuer: commonName=DigiCert SHA2 Secure Server CA/organizationName=DigiCert Inc/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
```

5.9 TLS certificate

Severity: Information
Confidence: Certain

Issue details:

The server presented a valid, trusted TLS certificate. This issue is purely informational. The server presented the following certificates:

Issue background:

TLS (or SSL) helps to protect the confidentiality and integrity of information in transit between the browser and server, and to provide authentication of the server's identity. To serve this purpose, the server must present an TLS certificate that is valid for the server's hostname, is issued by a trusted authority and is valid for the current date. If any one of these requirements is not met, TLS connections to the server will not provide the full protection for which TLS is designed.

Server certificate

Issued to: *.azurewebsites.net, *.scm.azurewebsites.net, *.azure-mobile.net, *.scm.azure-mobile.net, *.sso.azurewebsites.net
Issued by: Microsoft Azure TLS Issuing CA 01



Valid from: Tue Mar 15 00:09:55 IST 2022

Valid to: Fri Mar 10 00:09:55 IST 2023

Certificate chain #1

Issued to: Microsoft Azure TLS Issuing CA 01

Issued by: DigiCert Global Root G2

Valid from: Wed Jul 29 18:00:00 IST 2020

Valid to: Fri Jun 28 05:29:59 IST 2024

Certificate chain #2

Issued to: DigiCert Global Root G2

Issued by: DigiCert Global Root G2

Valid from: Thu Aug 01 17:30:00 IST 2013

Valid to: Fri Jan 15 17:30:00 IST 2038

Server certificate

Issued to: admin.smartmedic-testing.com

Issued by: GeoTrust Global TLS RSA4096 SHA256 2022 CA1

Valid from: Wed Jun 15 05:30:00 IST 2022

Valid to: Fri Dec 16 05:29:59 IST 2022

Certificate chain #1

Issued to: GeoTrust Global TLS RSA4096 SHA256 2022 CA1

Issued by: DigiCert Global Root CA

Valid from: Wed May 04 05:30:00 IST 2022

Valid to: Mon Nov 10 05:29:59 IST 2031

Certificate chain #2

Issued to: DigiCert Global Root CA

Issued by: DigiCert Global Root CA

Valid from: Fri Nov 10 05:30:00 IST 2006

Valid to: Mon Nov 10 05:30:00 IST 2031

Server certificate

Issued to: api.smartmedic-testing.com

Issued by: GeoTrust Global TLS RSA4096 SHA256 2022 CA1

Valid from: Thu Jun 16 05:30:00 IST 2022

Valid to: Sat Dec 17 05:29:59 IST 2022

Certificate chain #1



Issued to: GeoTrust Global TLS RSA4096 SHA256 2022 CA1

Issued by: DigiCert Global Root CA

Valid from: Wed May 04 05:30:00 IST 2022

Valid to: Mon Nov 10 05:29:59 IST 2031

Certificate chain #2

Issued to: DigiCert Global Root CA

Issued by: DigiCert Global Root CA

Valid from: Fri Nov 10 05:30:00 IST 2006

Valid to: Mon Nov 10 05:30:00 IST 2031

E. Annexure B

For the reference of the result where the vulnerability was not found, refer the annexure B which contains the list of all the vulnerabilities which were tested by GSL test scanners.

The below list depicts all the tests carried by the scanner on the application.

Name	Typical severity	Type index
OS command injection	High	0x00100100
SQL injection	High	0x00100200
SQL injection (second order)	High	0x00100210
ASP.NET tracing enabled	High	0x00100280
File path traversal	High	0x00100300
XML external entity injection	High	0x00100400
LDAP injection	High	0x00100500
XPath injection	High	0x00100600
XML injection	Medium	0x00100700
ASP.NET debugging enabled	Medium	0x00100800
HTTP PUT method is enabled	High	0x00100900
Out-of-band resource load (HTTP)	High	0x00100a00
File path manipulation	High	0x00100b00
PHP code injection	High	0x00100c00
Server-side JavaScript code injection	High	0x00100d00
Perl code injection	High	0x00100e00
Ruby code injection	High	0x00100f00
Python code injection	High	0x00100f10
Expression Language injection	High	0x00100f20
Unidentified code injection	High	0x00101000
Server-side template injection	High	0x00101080
SSI injection	High	0x00101100
Cross-site scripting (stored)	High	0x00200100



Name	Typical severity	Type index
HTTP request smuggling	High	0x00200140
Web cache poisoning	High	0x00200180
HTTP response header injection	High	0x00200200
Cross-site scripting (reflected)	High	0x00200300
Client-side template injection	High	0x00200308
Cross-site scripting (DOM-based)	High	0x00200310
Cross-site scripting (reflected DOM-based)	High	0x00200311
Cross-site scripting (stored DOM-based)	High	0x00200312
JavaScript injection (DOM-based)	High	0x00200320
JavaScript injection (reflected DOM-based)	High	0x00200321
JavaScript injection (stored DOM-based)	High	0x00200322
Path-relative style sheet import	Information	0x00200328
Client-side SQL injection (DOM-based)	High	0x00200330
Client-side SQL injection (reflected DOM-based)	High	0x00200331
Client-side SQL injection (stored DOM-based)	High	0x00200332
WebSocket URL poisoning (DOM-based)	High	0x00200340
WebSocket URL poisoning (reflected DOM-based)	High	0x00200341
WebSocket URL poisoning (stored DOM-based)	High	0x00200342
Local file path manipulation (DOM-based)	High	0x00200350
Local file path manipulation (reflected DOM-based)	High	0x00200351
Local file path manipulation (stored DOM-based)	High	0x00200352
Client-side XPath injection (DOM-based)	Low	0x00200360



Name	Typical severity	Type index
Client-side XPath injection (reflected DOM-based)	Low	0x00200361
Client-side XPath injection (stored DOM-based)	Low	0x00200362
Client-side JSON injection (DOM-based)	Low	0x00200370
Client-side JSON injection (reflected DOM-based)	Low	0x00200371
Client-side JSON injection (stored DOM-based)	Low	0x00200372
Flash cross-domain policy	High	0x00200400
Silverlight cross-domain policy	High	0x00200500
Cross-origin resource sharing	Information	0x00200600
Cross-origin resource sharing: arbitrary origin trusted	High	0x00200601
Cross-origin resource sharing: unencrypted origin trusted	Low	0x00200602
Cross-origin resource sharing: all subdomains trusted	Low	0x00200603
Cross-site request forgery	Medium	0x00200700
SMTP header injection	Medium	0x00200800
JWT signature not verified	High	0x00200900
JWT none algorithm supported	High	0x00200901
JWT self-signed JWK header supported	High	0x00200902
JWT weak HMAC secret	High	0x00200903
JWT arbitrary jku header supported	High	0x00200904
JWT arbitrary x5u header supported	High	0x00200905
Cleartext submission of password	High	0x00300100



Name	Typical severity	Type index
External service interaction (SMTP)	Information	0x00300220
Referer-dependent response	Information	0x00400100
Spoofable client IP address	Information	0x00400110
User agent-dependent response	Information	0x00400120
Password returned in later response	Medium	0x00400200
Password submitted using GET method	Low	0x00400300
Password returned in URL query string	Low	0x00400400
SQL statement in request parameter	Medium	0x00400480
Cross-domain POST	Information	0x00400500
ASP.NET ViewState without MAC enabled	High	0x00400600
XML entity expansion	Medium	0x00400700
Long redirection response	Information	0x00400800
Serialized object in HTTP message	High	0x00400900
Duplicate cookies set	Information	0x00400a00
Input returned in response (stored)	Information	0x00400b00
Input returned in response (reflected)	Information	0x00400c00
Suspicious input transformation (reflected)	Information	0x00400d00
Suspicious input transformation (stored)	Information	0x00400e00
Request URL override	Information	0x00400f00
Vulnerable JavaScript dependency	Low	0x00500080
Open redirection (reflected)	Low	0x00500100
Open redirection (stored)	Medium	0x00500101



Name	Typical severity	Type index
Open redirection (DOM-based)	Low	0x00500110
Open redirection (reflected DOM-based)	Low	0x00500111
Open redirection (stored DOM-based)	Medium	0x00500112
TLS cookie without secure flag set	Medium	0x00500200
Cookie scoped to parent domain	Low	0x00500300
Cross-domain Referer leakage	Information	0x00500400
Cross-domain script include	Information	0x00500500
Cookie without HttpOnly flag set	Low	0x00500600
Session token in URL	Medium	0x00500700
Password field with autocomplete enabled	Low	0x00500800
Password value set in cookie	Medium	0x00500900
File upload functionality	Information	0x00500980
Frameable response (potential Clickjacking)	Information	0x005009a0
Browser cross-site scripting filter disabled	Information	0x005009b0
HTTP TRACE method is enabled	Information	0x00500a00
Cookie manipulation (DOM-based)	Low	0x00500b00
Cookie manipulation (reflected DOM-based)	Low	0x00500b01
Cookie manipulation (stored DOM-based)	Low	0x00500b02
Ajax request header manipulation (DOM-based)	Low	0x00500c00
Ajax request header manipulation (reflected DOM-based)	Low	0x00500c01
Ajax request header manipulation (stored DOM-based)	Low	0x00500c02



Name	Typical severity	Type index
HTML5 web message manipulation (DOM-based)	Information	0x00500e00
HTML5 web message manipulation (reflected DOM-based)	Information	0x00500e01
HTML5 web message manipulation (stored DOM-based)	Information	0x00500e02
HTML5 storage manipulation (DOM-based)	Information	0x00500f00
HTML5 storage manipulation (reflected DOM-based)	Information	0x00500f01
HTML5 storage manipulation (stored DOM-based)	Information	0x00500f02
Link manipulation (DOM-based)	Low	0x00501000
Link manipulation (reflected DOM-based)	Low	0x00501001
Link manipulation (stored DOM-based)	Low	0x00501002
Link manipulation (reflected)	Information	0x00501003
Link manipulation (stored)	Information	0x00501004
Document domain manipulation (DOM-based)	Medium	0x00501100
Document domain manipulation (reflected DOM-based)	Medium	0x00501101
Document domain manipulation (stored DOM-based)	Medium	0x00501102
DOM data manipulation (DOM-based)	Information	0x00501200
DOM data manipulation (reflected DOM-based)	Information	0x00501201
DOM data manipulation (stored DOM-based)	Information	0x00501202
CSS injection (reflected)	Medium	0x00501300
CSS injection (stored)	Medium	0x00501301
Client-side HTTP parameter pollution (reflected)	Low	0x00501400
Client-side HTTP parameter pollution (stored)	Low	0x00501401
Form action hijacking (reflected)	Medium	0x00501500
Form action hijacking (stored)	Medium	0x00501501



Database connection string disclosed	Medium	0x00600080
Source code disclosure	Low	0x006000b0
Backup file	Information	0x006000d8
Directory listing	Information	0x00600100
Email addresses disclosed	Information	0x00600200
Private IP addresses disclosed	Information	0x00600300
Social security numbers disclosed	Information	0x00600400
Credit card numbers disclosed	Information	0x00600500
Private key disclosed	Information	0x00600550
Robots.txt file	Information	0x00600600
Json Web Key Set disclosed	Information	0x00600700
JWT private key disclosed	Information	0x00600800
Cacheable HTTPS response	Information	0x00700100
Base64-encoded data in parameter	Information	0x00700200
Multiple content types specified	Information	0x00800100
HTML does not specify charset	Information	0x00800200
HTML uses unrecognized charset	Information	0x00800300
Content type incorrectly stated	Low	0x00800400
Content type is not specified	Information	0x00800500
TLS certificate	Medium	0x01000100
Unencrypted communications	Low	0x01000200
Strict transport security not enforced	Low	0x01000300
Mixed content	Information	0x01000400
Hidden HTTP 2	Information	0x01000500
Extension generated issue	Information	0x08000000



Name	Typical severity	Type index
File upload functionality	Information	0x00500980
Frameable response (potential Clickjacking)	Information	0x005009a0
Browser cross-site scripting filter disabled	Information	0x005009b0
HTTP TRACE method is enabled	Information	0x00500a00
Cookie manipulation (DOM-based)	Low	0x00500b00
Cookie manipulation (reflected DOM-based)	Low	0x00500b01
Cookie manipulation (stored DOM-based)	Low	0x00500b02
Ajax request header manipulation (DOM-based)	Low	0x00500c00
Ajax request header manipulation (reflected DOM-based)	Low	0x00500c01
Ajax request header manipulation (stored DOM-based)	Low	0x00500c02
Denial of service (DOM-based)	Information	0x00500d00
Denial of service (reflected DOM-based)	Information	0x00500d01
Denial of service (stored DOM-based)	Low	0x00500d02
HTML5 web message manipulation (DOM-based)	Information	0x00500e00
HTML5 web message manipulation (reflected DOM-based)	Information	0x00500e01
HTML5 web message manipulation (stored DOM-based)	Information	0x00500e02
HTML5 storage manipulation (DOM-based)	Information	0x00500f00
HTML5 storage manipulation (reflected DOM-based)	Information	0x00500f01
HTML5 storage manipulation (stored DOM-based)	Information	0x00500f02
Link manipulation (DOM-based)	Low	0x00501000
Link manipulation (reflected DOM-based)	Low	0x00501001
Link manipulation (stored DOM-based)	Low	0x00501002
Link manipulation (reflected)	Information	0x00501003
Link manipulation (stored)	Information	0x00501004
Document domain manipulation (DOM-based)	Medium	0x00501100



← → ⌛ 🔒 https://admin.smartmedic-testing.com

 Sandbox iframe?
allow-scripts allow-forms



admin.smartmedic-testing.com refused to connect.

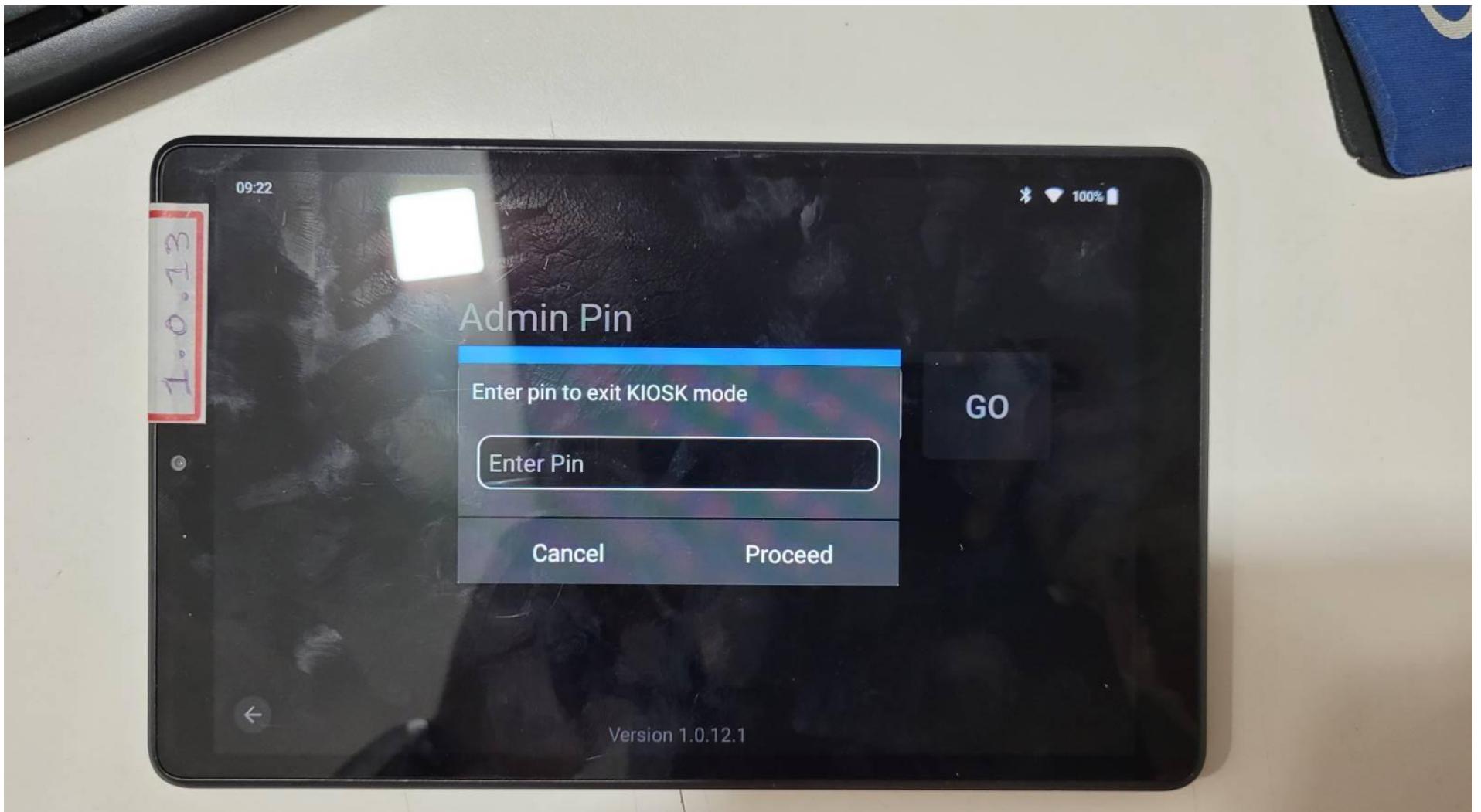
← → ⌛ 🔒 https://smartmedic-testing.com

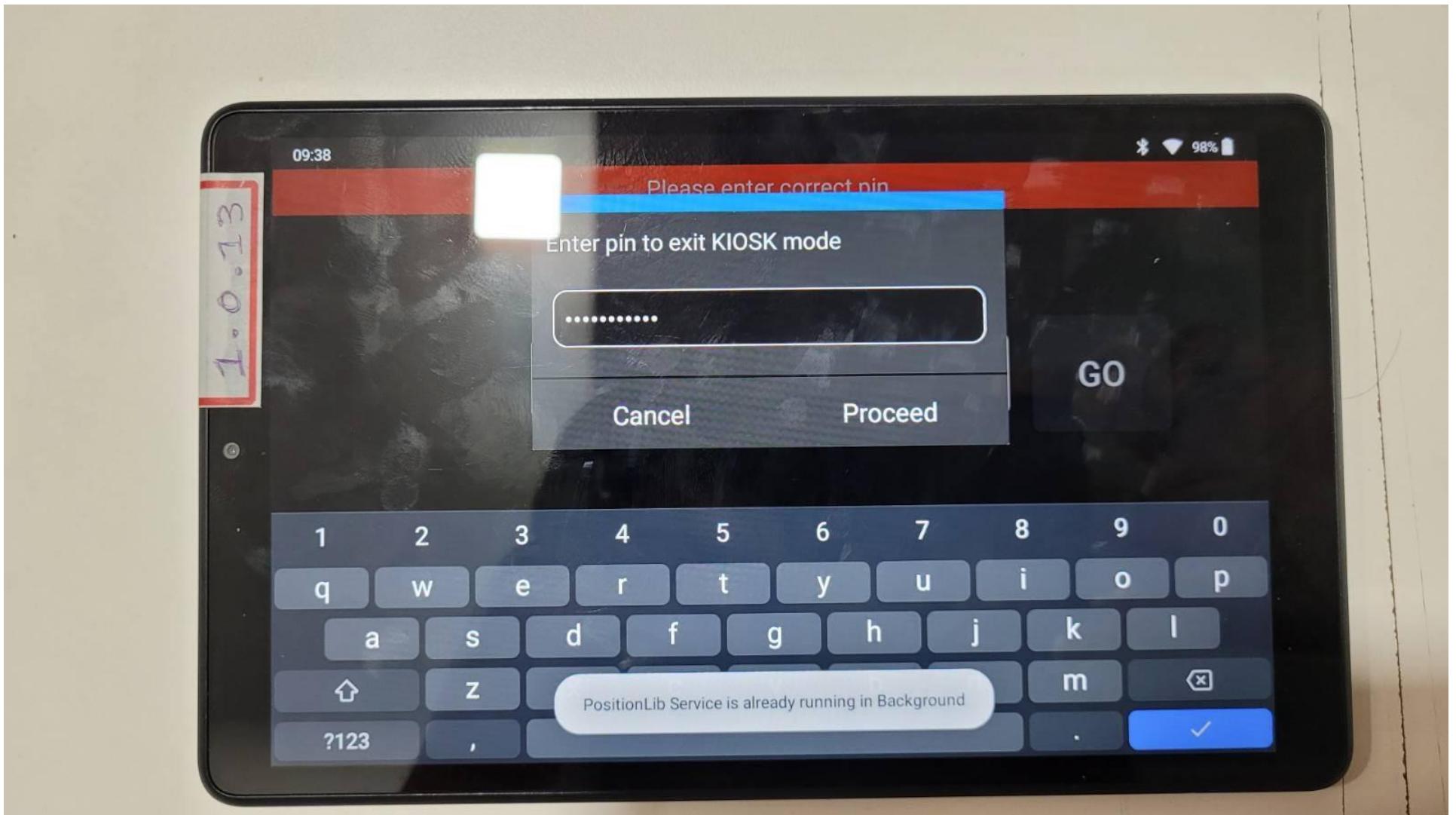
 Sandbox iframe?

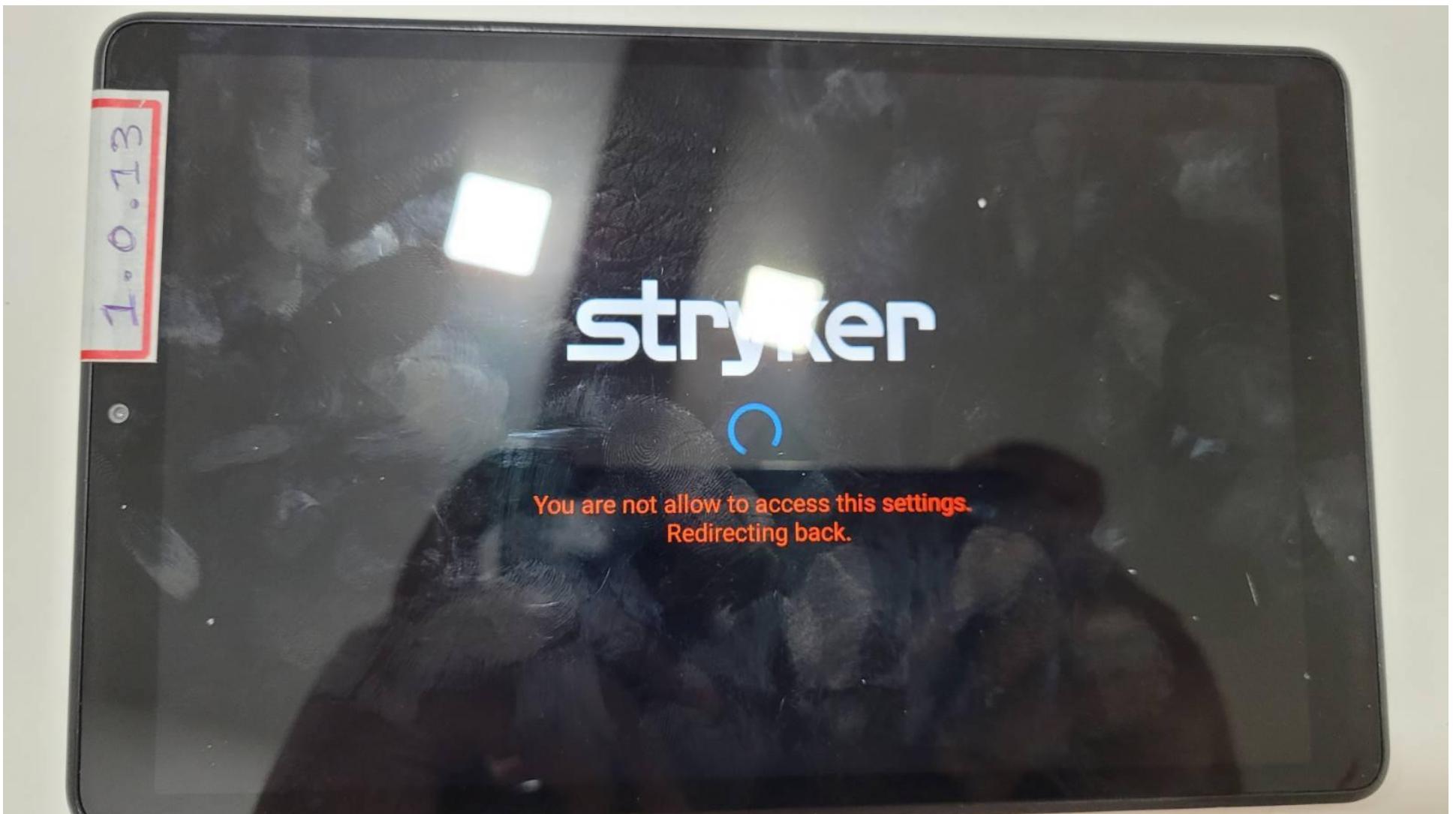


smartmedic-testing.com refused to connect.



F. Annexure B.1



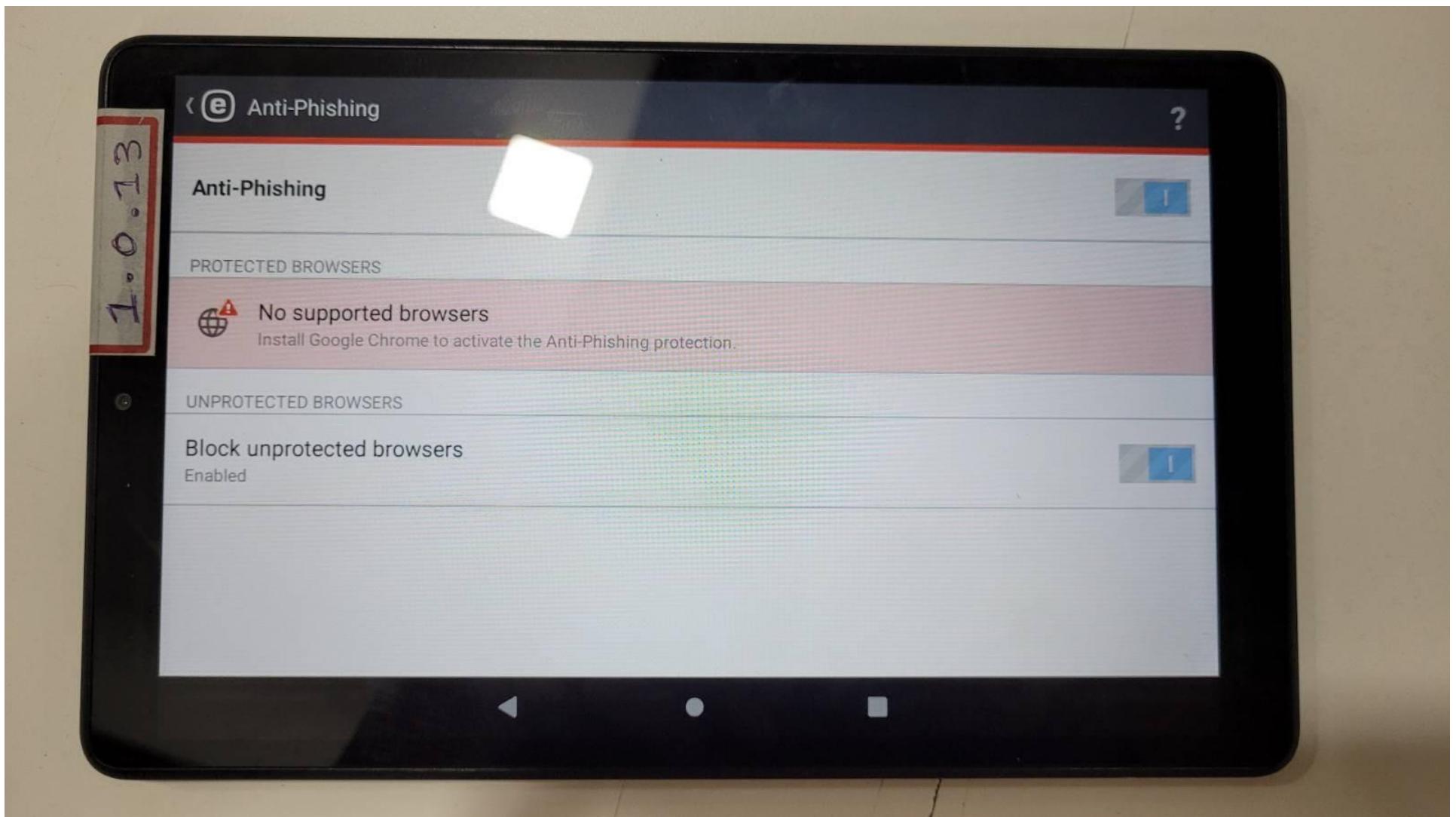


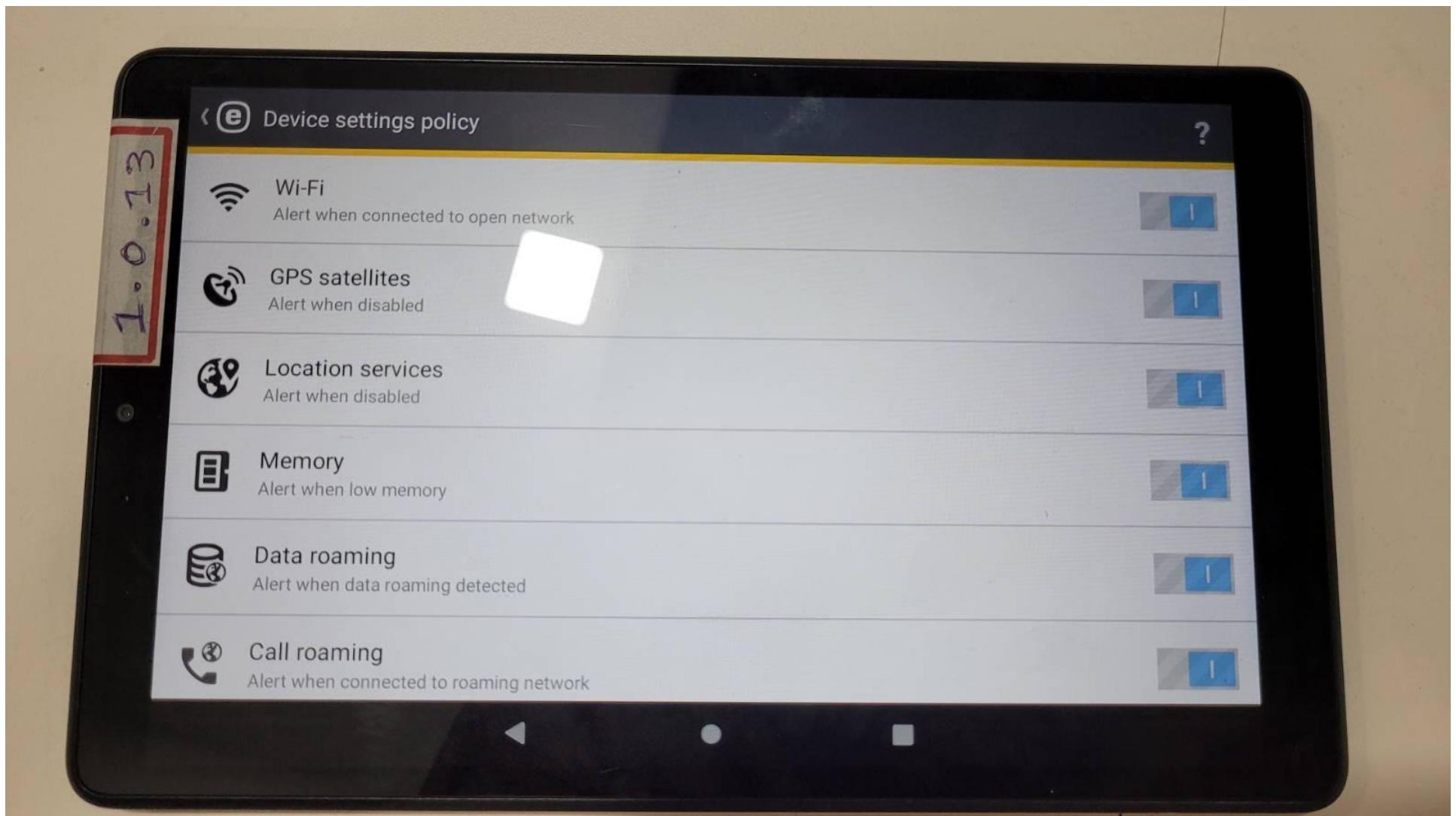
CO

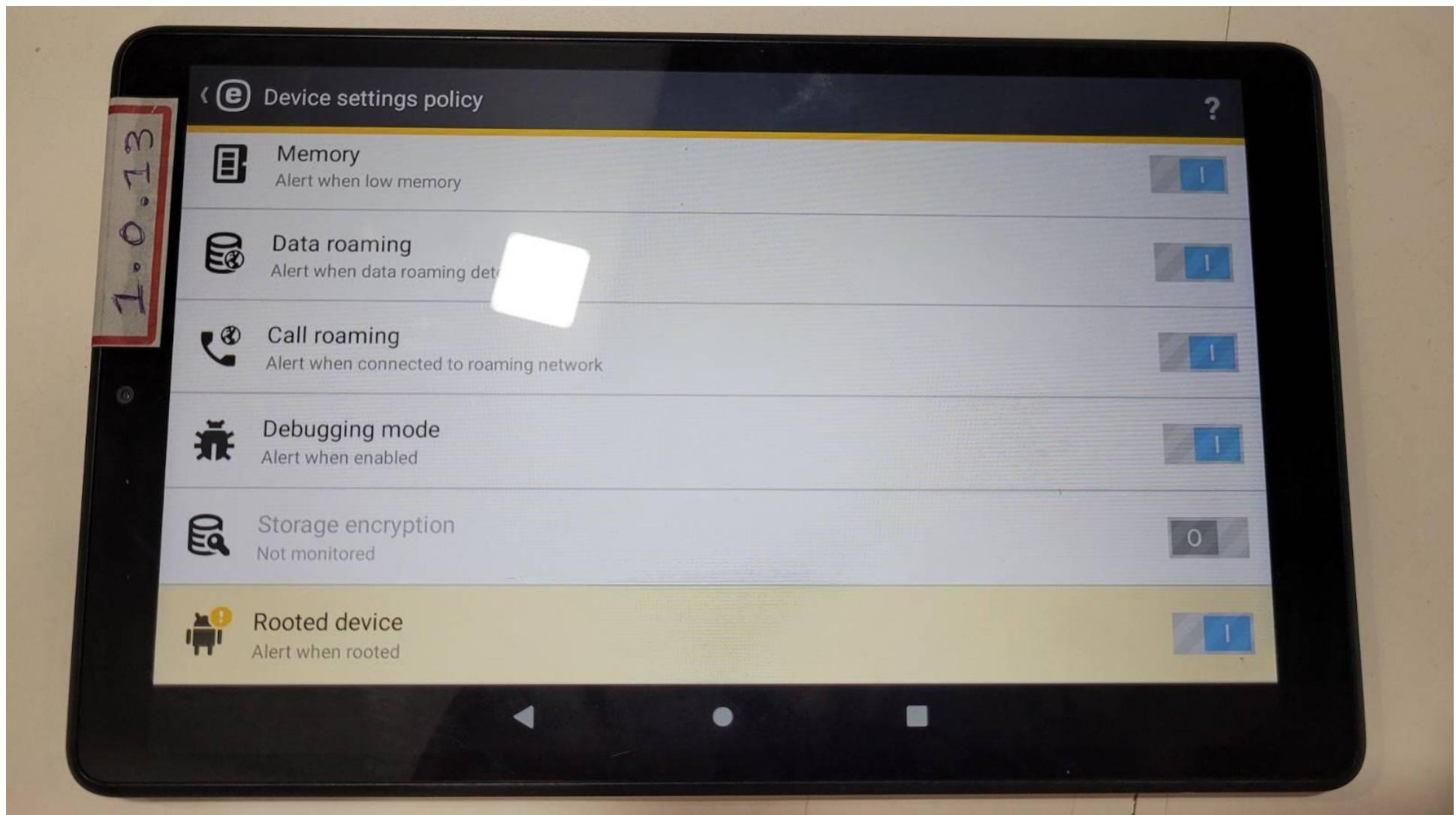


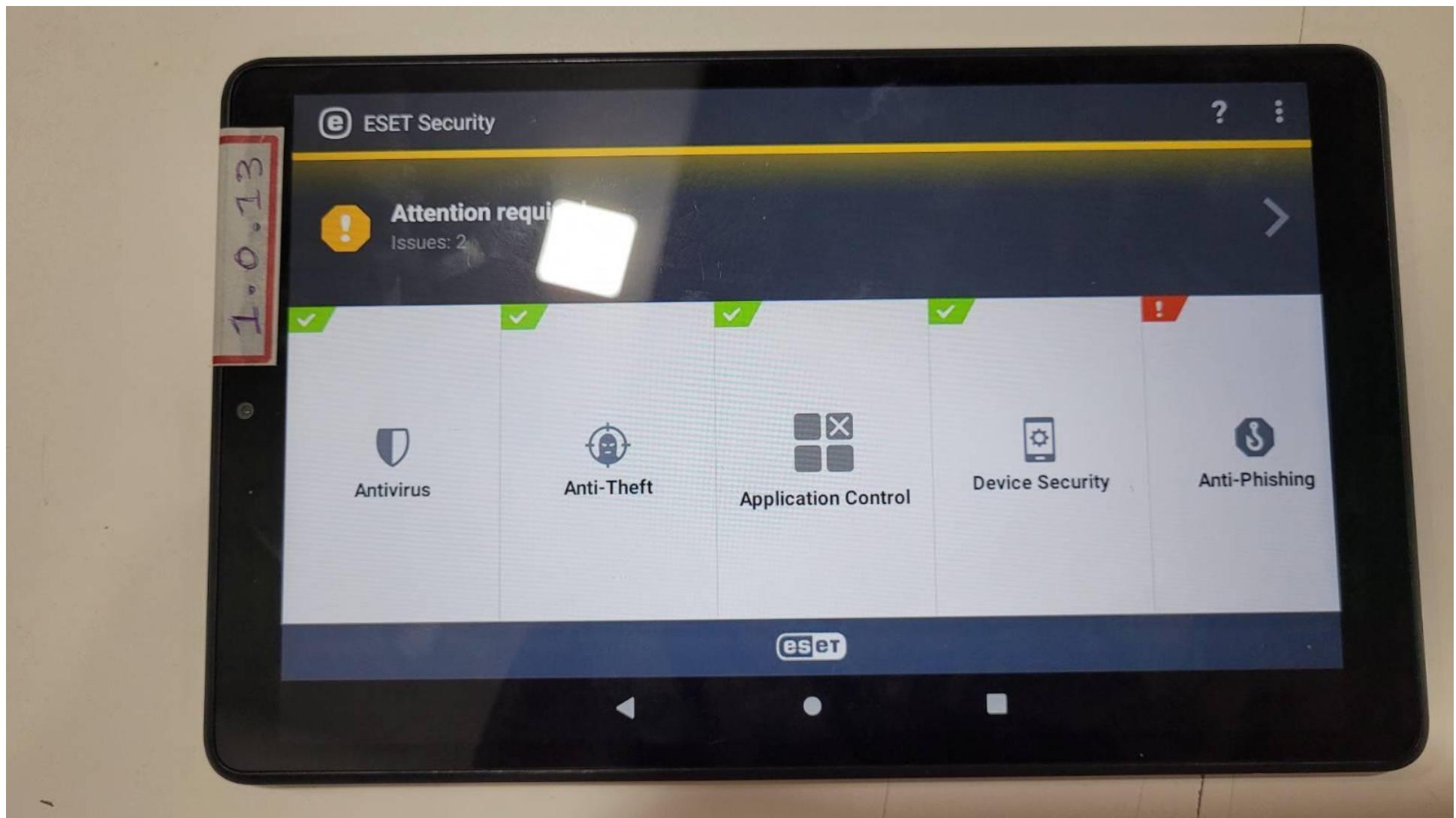


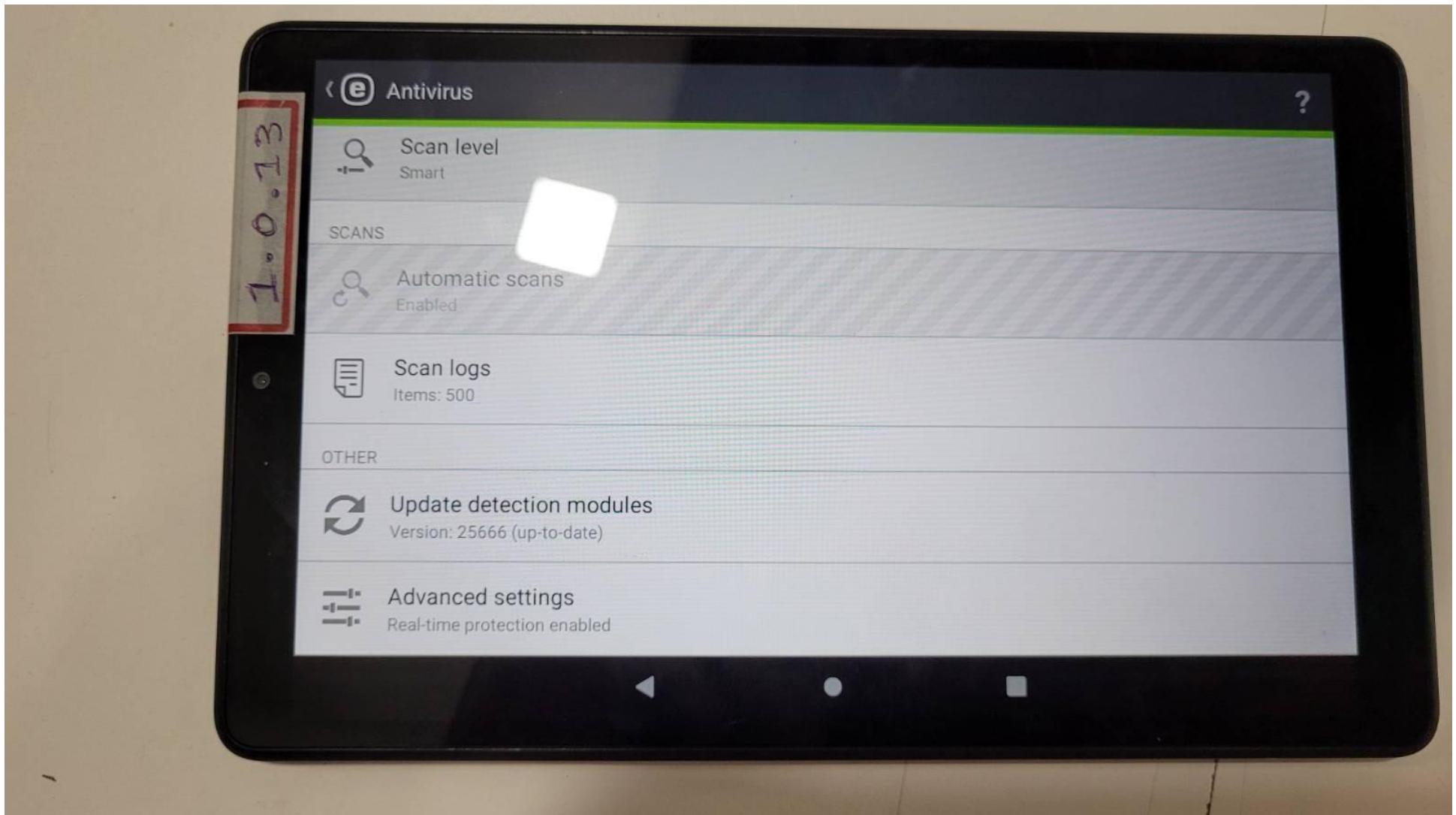


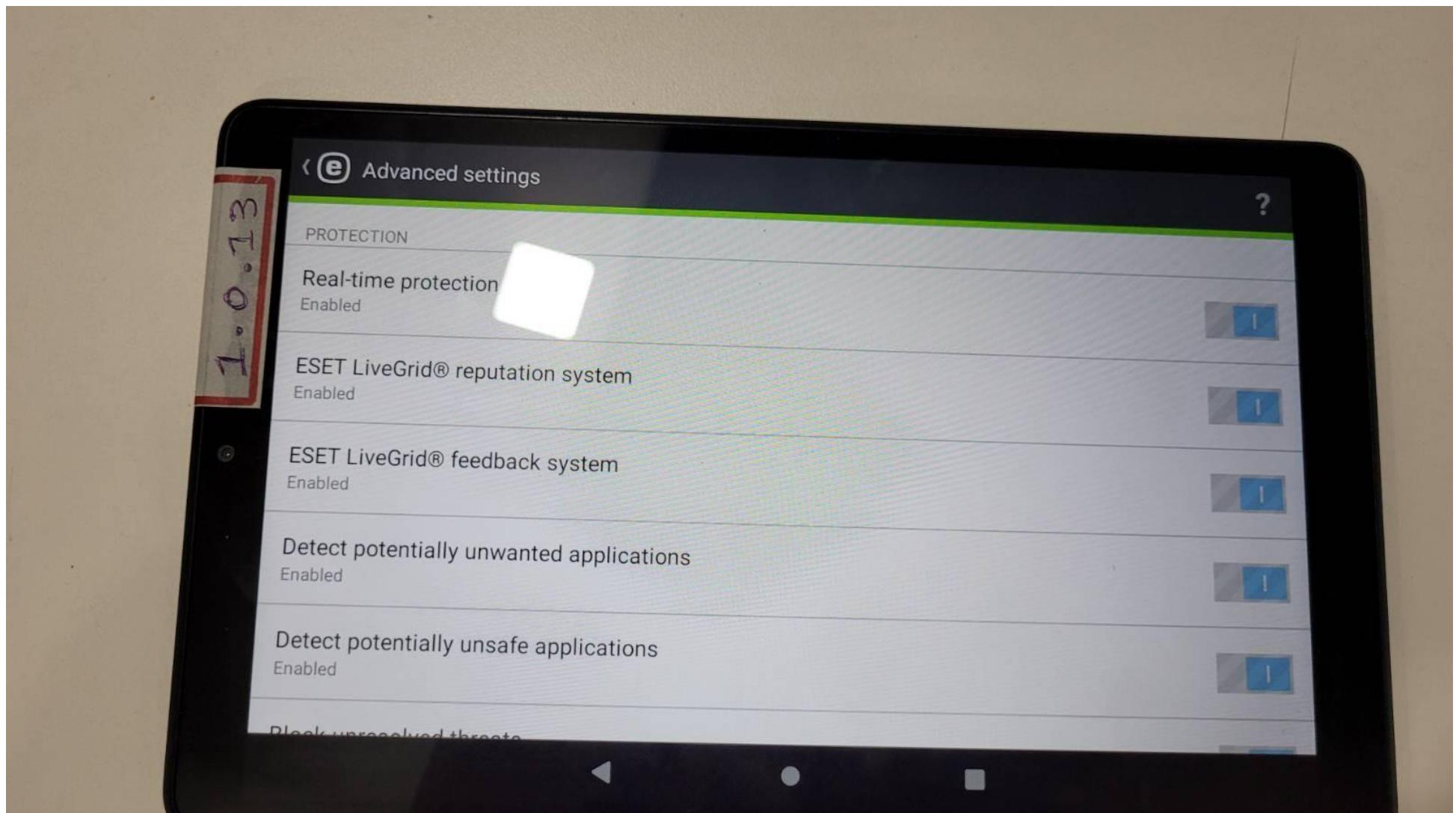


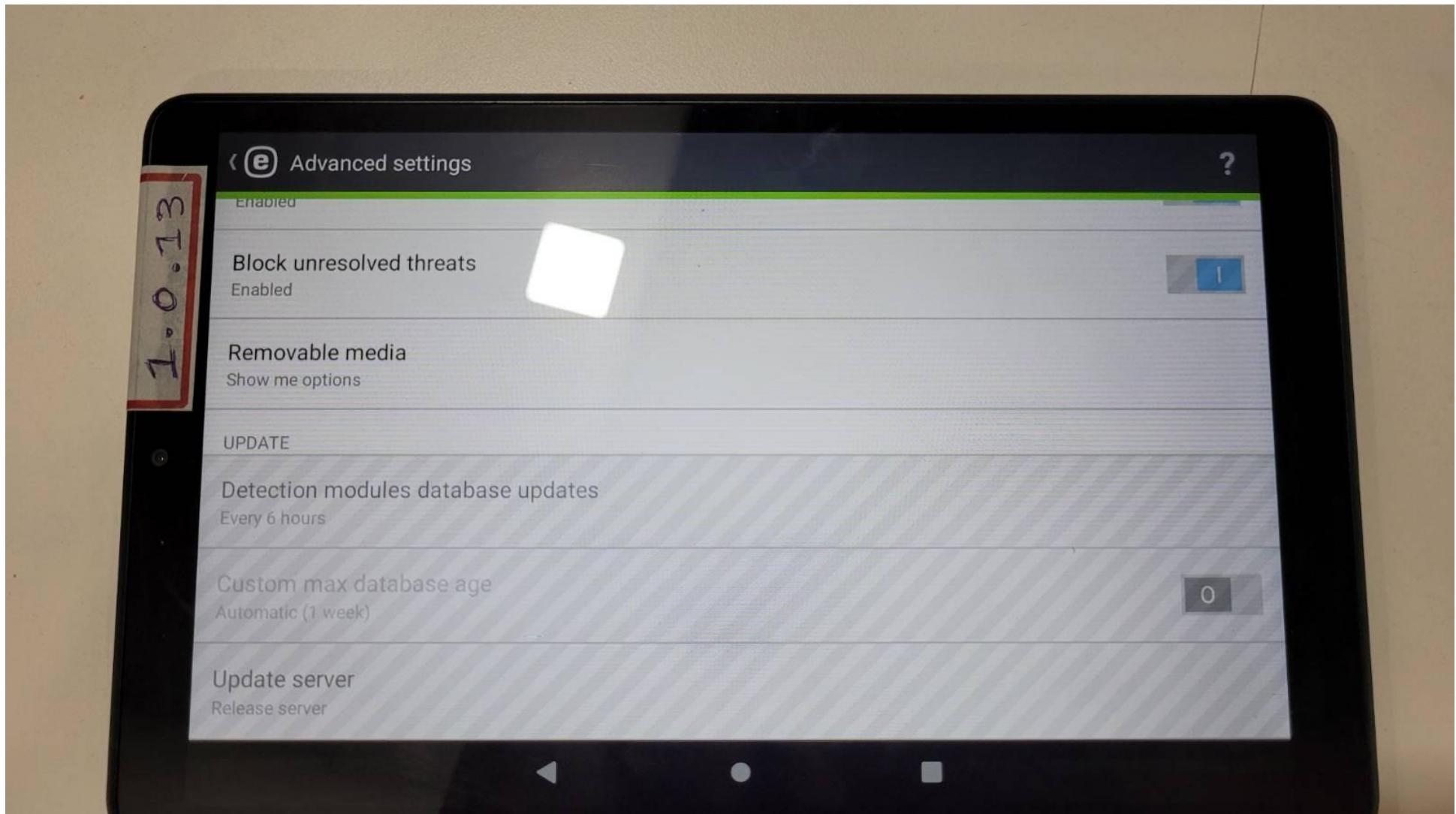


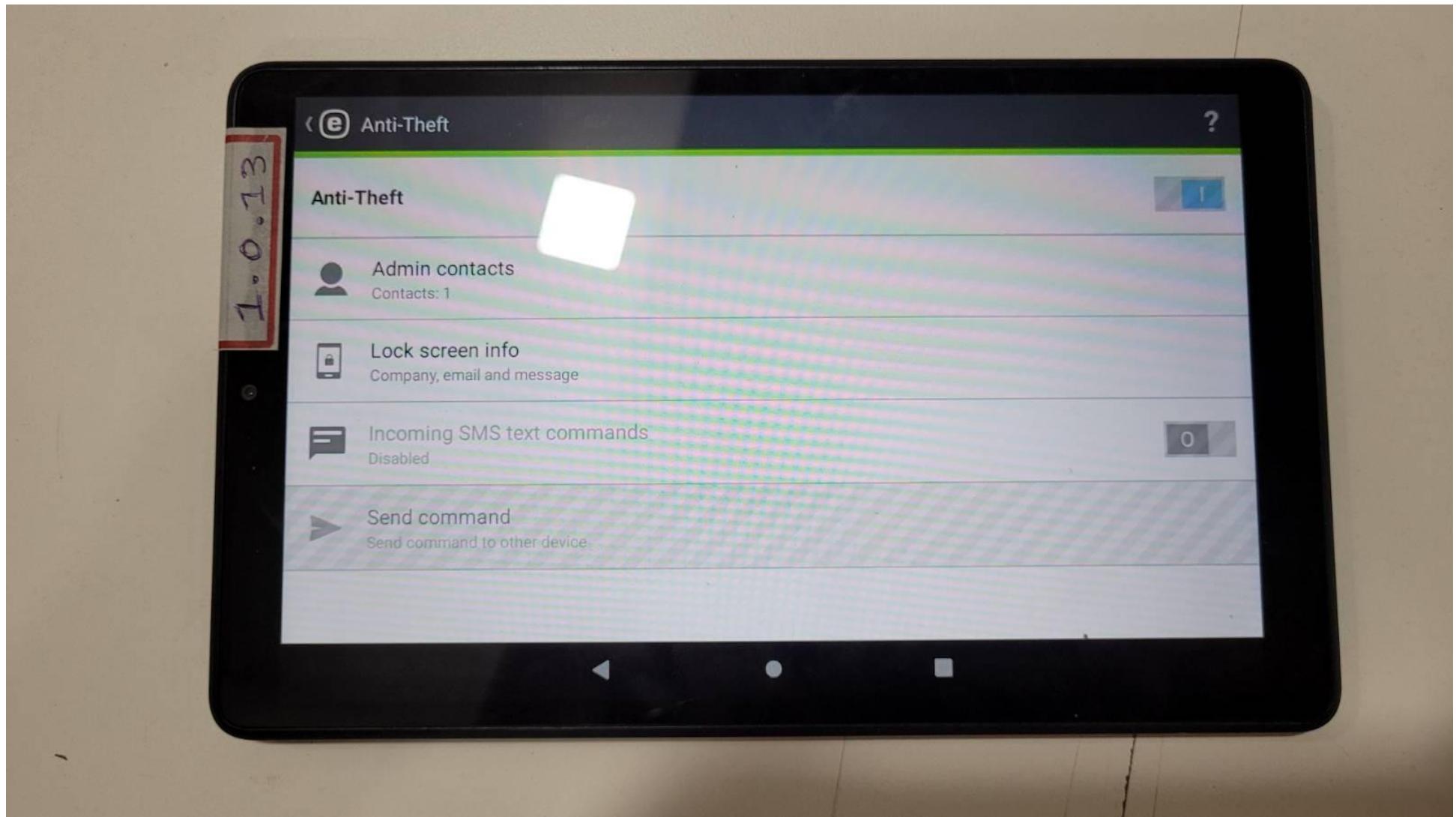


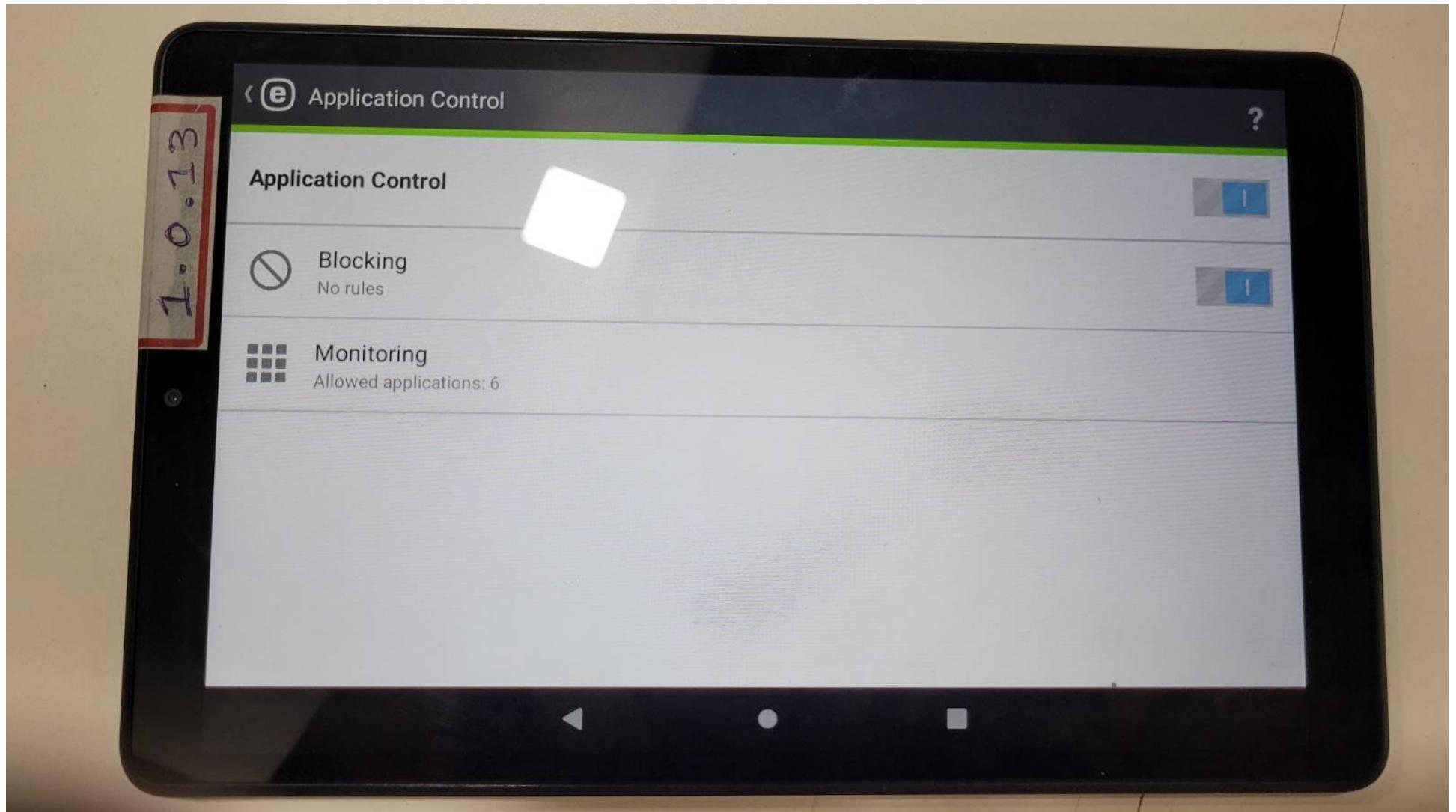


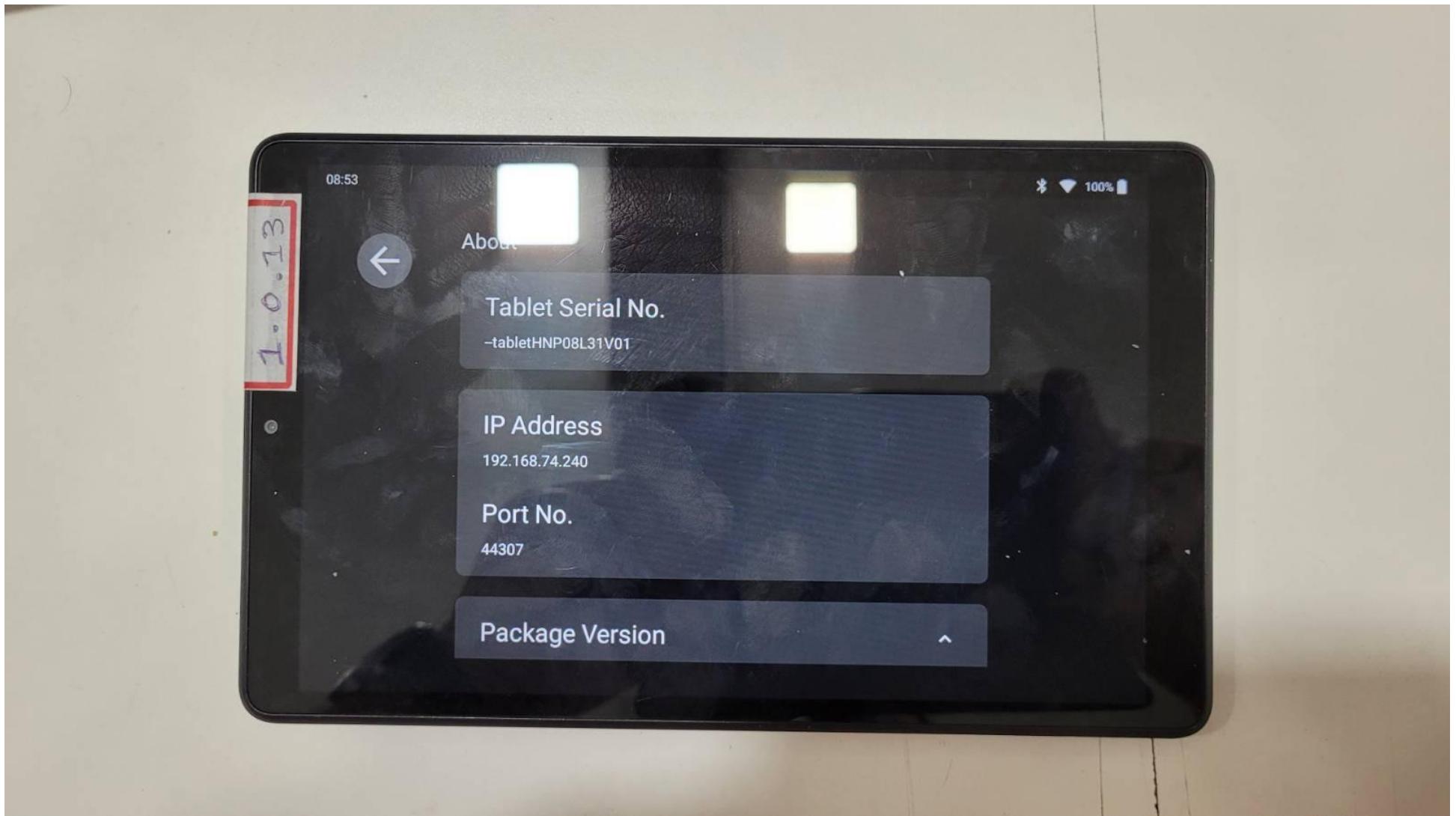


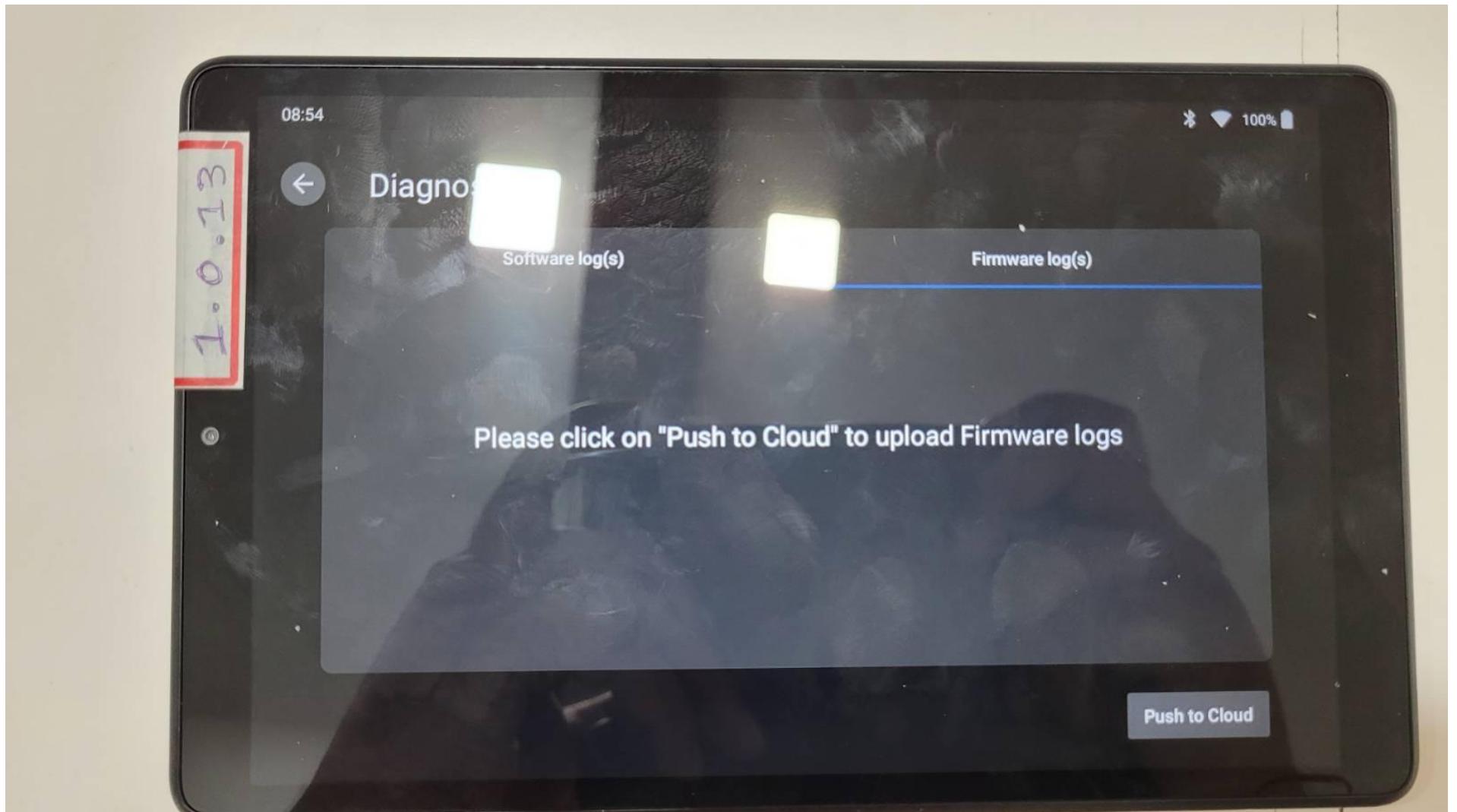


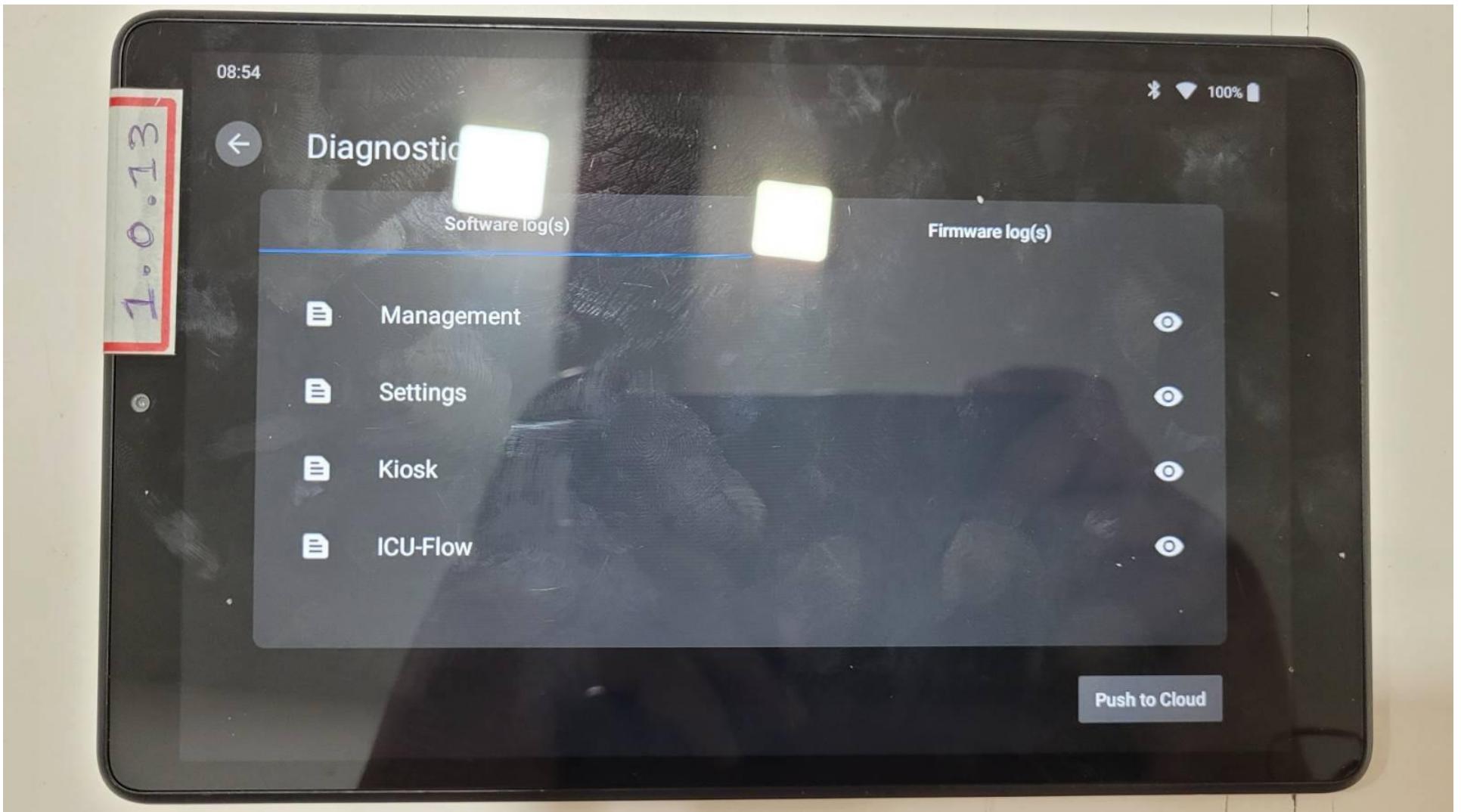












[Info] AndroidManifest Adb Backup Checking:

This app has disabled Adb Backup.

[Info] <Command> Runtime Command Checking:

This app is not using critical function 'Runtime.getRuntime().exec("...")'.

[Info] <Database> SQLiteDatabase Transaction Deprecated Checking:

Ignore checking "SQLiteDatabase:beginTransactionNonExclusive" because your set minSdk >= 11.

[Info] <Database> Android SQLite Databases Encryption (SQLite Encryption Extension (SEE)):



This app is "NOT" using SQLite Encryption Extension (SEE) on Android (<http://www.sqlite.org/android>) to encrypt or decrypt databases.

[Info] <Database> Android SQLite Databases Encryption (SQLCipher):
This app is "NOT" using SQLCipher(<http://sqlcipher.net/>) to encrypt or decrypt databases.

[Info] <Debug> Android Debug Mode Checking:
DEBUG mode is OFF(android:debuggable="false") in AndroidManifest.xml.

[Info] Dynamic Code Loading:
No dynamic code loading(DexClassLoader) found.

[Info] <#BID 64208, CVE-2013-6271#> Fragment Vulnerability Checking:
Did not detect the vulnerability of "Fragment" dynamically loading into "PreferenceActivity" or "SherlockPreferenceActivity"

[Info] <Framework> Framework - MonoDroid:
This app is NOT using MonoDroid Framework (<http://xamarin.com/android>).

[Info] <Hacker> Base64 String Encryption:
No encoded Base64 String or URLs found.

[Info] <Database><Hacker> Key for Android SQLite Databases Encryption:
Did not find using the symmetric key(PRAGMA key) to encrypt the SQLite databases (It's still possible that it might use but we did not find out).

[Info] <Hacker> APK Installing Source Checking:
Did not detect this app checks for APK installer sources.

[Info] <KeyStore><Hacker> KeyStore File Location:
Did not find any possible BKS keystores or certificate keystore file (Notice: It does not mean this app does not use keysotre):

[Info] <Hacker> Code Setting Preventing Screenshot Capturing:
Did not detect this app has code setting preventing screenshot capturing.

[Info] HttpURLConnection Android Bug Checking:
Ignore checking "http.keepAlive" because you're not using "HttpURLConnection" and min_Sdk > 8.

[Info] <KeyStore> KeyStore Type Checking:
KeyStore 'BKS' type check OK

[Info] Google Cloud Messaging Suggestion:
Nothing to suggest.

[Info] <#CVE-2013-4787#> Master Key Type I Vulnerability:
No Master Key Type I Vulnerability in this APK.

[Info] App Sandbox Permission Checking:
No security issues "MODE_WORLD_READABLE" or "MODE_WORLD_WRITEABLE" found on 'openOrCreateDatabase' or 'openOrCreateDatabase2' or 'getDir' or 'getSharedPreferences' or 'openFileOutput'

[Info] Native Library Loading Checking:
No native library loaded.

[Info] AndroidManifest Dangerous ProtectionLevel of Permission Checking:



No "dangerous" protection level customized permission found (AndroidManifest.xml).

[Info] AndroidManifest PermissionGroup Checking:
 PermissionGroup in permission tag of AndroidManifest sets correctly.

[Info] AndroidManifest "intent-filter" Settings Checking:
 "intent-filter" of AndroidManifest.xml check OK.

[Info] AndroidManifest Normal ProtectionLevel of Permission Checking:
 No default or "normal" protection level customized permission found (AndroidManifest.xml).

[Info] <#CVE-2013-6272#> AndroidManifest Exported Lost Prefix Checking:
 No exported components that forgot to add "android:" prefix.

[Info] AndroidManifest ContentProvider Exported Checking:
 No exported "ContentProvider" found (AndroidManifest.xml).

[Info] <Sensitive_Information> Getting ANDROID_ID:
 Did not detect this app is getting the 64-bit number "Settings.Secure.ANDROID_ID".

[Info] Codes for Sending SMS:
 Did not detect this app has code for sending SMS messages (sendDataMessage, sendMultipartTextMessage or sendTextMessage).

[Info] <System> AndroidManifest sharedUserId Checking:
 This app does not use "android.uid.system" sharedUserId.

[Info] <SSL_Security> SSL Implementation Checking (Verifying Host Name in Custom Classes):
 Self-defined HOSTNAME VERIFIER checking OK.

[Info] <SSL_Security> SSL Implementation Checking (Verifying Host Name in Fields):
 Critical vulnerability "ALLOW_ALL_HOSTNAME_VERIFIER" field setting or "AllowAllHostnameVerifier" class instance not found.

[Info] <SSL_Security> SSL Implementation Checking (Insecure component):
 Did not detect SSLSocketFactory by insecure method "getInsecure".

[Info] <SSL_Security> SSL Implementation Checking (HttpHost):
 DEFAULT_SCHEME_NAME for HttpHost check: OK

[Info] <SSL_Security> SSL Implementation Checking (WebViewClient for WebView):
 Did not detect critical usage of "WebViewClient"(MITM Vulnerability).

[Info] <SSL_Security> SSL Certificate Verification Checking:
 Did not find vulnerable X509Certificate code.

[Info] Unnecessary Permission Checking:
 Permission 'android.permission.ACCESS_MOCK_LOCATION' sets correctly.

[Info] Accessing the Internet Checking:
 This app is using the Internet via HTTP protocol.

[Info] AndroidManifest System Use Permission Checking:
 No system-level critical use-permission found.

[Info] <WebView> WebView Local File Access Attacks Checking:
 Did not find potentially critical local file access settings.



[Info] <WebView> WebView Potential XSS Attacks Checking:
Did not detect "setJavaScriptEnabled(true)" in WebView.

[Info] <WebView><Remote Code Execution><#CVE-2013-4710#> WebView RCE Vulnerability Checking:
WebView addJavascriptInterface vulnerabilities not found.

G. Annexure C

For reference of functional test result refer the document which illustrate the functional test case results for Smart Medic Solution provided by the development team.

CONFIDENTIAL



** End of the document



INDIA

B/81, Corporate House,
Judges Bungalow Road, Bodakdev,
Ahmedabad - 380054. India

THE NETHERLAND

Maria Montessorilaan 3,
2719 DB Zoetermeer,
The Netherlands

www.gsecurelabs.com



Confidentiality Clause:

This document and any files with it are for the sole use of the intended recipient(s) and may contain confidential and privileged information. If you are not the intended recipient, please destroy all copies of the document. Any unauthorized review, use, disclosure, dissemination, forwarding, printing or copying of this document or any action taken in reliance on this document is strictly prohibited and may be unlawful.

Copyright © Gateway Group

